Package 'cond'

April 16, 2018

Version 1.2-3.1

Date 2014-06-27

Title Approximate Conditional Inference for Logistic and Loglinear Models

Author S original by Alessandra R. Brazzale <alessandra.brazzale@unipd.it>. R port by Alessandra R. Brazzale <alessandra.brazzale@unipd.it>, following earlier work by Douglas Bates.

Maintainer Alessandra R. Brazzale <alessandra.brazzale@unipd.it>

Depends R (>= 3.0.0), statmod, survival

Suggests boot, csampling, marg, nlreg

Description Higher order likelihood-based inference for logistic and loglinear models.

License GPL (>= 2) | file LICENCE

URL https://www.r-project.org, http://statwww.epfl.ch/AA/

LazyLoad yes

LazyData yes

NeedsCompilation no

Repository CRAN

Date/Publication 2018-04-16 15:01:42 UTC

R topics documented:

`cond-package` *Approximate conditional inference for logistic and loglinear models*

Description

Higher order likelihood-based inference for logistic and loglinear models

Details

Package:	cond
Version:	1.2-0
Date:	2009-10-03
Depends:	R (>= 2.6.0), statmod, survival
Suggests:	csampling, marg, nlreg
License:	GPL (>= 2)
URL:	http://www.r-project.org, http://statwww.epfl.ch/AA/
LazyLoad:	yes
LazyData:	yes

Index:

```
Functions:
=========
cond                    Approximate Conditional Inference - Generic
                        Function
cond.glm                Approximate Conditional Inference for Logistic
                        and Loglinear Models
cond.object             Approximate Conditional Inference Object
family.cond             Use family() on a "cond" object
family.summary.cond     Use family() on a "summary.cond" object
plot.cond               Generate Plots for an Approximate Conditional
                        Inference Object
print.summary.cond      Use print() on a "summary.cond" object
summary.cond            Summary Method for Objects of Class "cond"
```

```
Datasets:
========
aids                    AIDS Symptoms and AZT Use Data
airway                  Airway Data
babies                  Crying Babies Data
dormicum                Dormicum Data
fraudulent              Fraudulent Automobile Insurance Claims Data
fungal                  Fungal Infections Treatment Data
rabbits                 Rabbits Data
urine                   Urine Data
```

Further information is available in the following vignettes:

`Rnews-paper`	hoa: An R Package Bundle for Higher Order Likelihood Inference (source, pdf)

Author(s)

S original by Alessandra R. Brazzale <alessandra.brazzale@unimore.it>. R port by Alessandra R. Brazzale <alessandra.brazzale@unimore.it>, following earlier work by Douglas Bates.

Maintainer: Alessandra R. Brazzale <alessandra.brazzale@unimore.it>

`aids` *AIDS Symptoms and AZT Use Data*

Description

The `aids` data frame has 4 rows and 4 columns.

On February 15, 1991, the *New York Times* published the results of a study on the presence of AIDS symptoms and AZT use. The data were cross-classified according to the race of the patients.

Usage

```
data(aids)
```

Format

This data frame contains the following columns:

`yes` the number of patients with AIDS symptoms;

`no` the number of patients without AIDS symptoms;

`azt` an indicator variable for AZT use;

`race` an indicator variable for the race (`w`=white, `b`=black).

Source

The data were obtained from the *New York Times* (2/15/91).

Examples

```
data(aids)
summary(aids)
```

---

airway *Airway Data*

Description

The `airway` data frame has 35 rows and 6 columns.

Study to compare two devices (tracheal tube and laryngeal mask) used to secure airway in patients undergoing surgery. The response variable is the presence of a sore throat. Further information on age, sex, use of a lubricant, and duration of the surgery is available.

Usage

```
data(airway)
```

Format

This data frame contains the following columns:

`response` an indicator variable for sore throat (1=yes, 0=no);

`type` the type of airway used (1=tracheal tube, 0=laryngeal mask);

`age` the age of the patient (in years);

`sex` an indicator variable for sex (1=male, 0=female);

`lubricant` an indicator variable for lubricant use (1=yes, 0=no);

`duration` the duration of the surgery (in minutes).

Source

The data were obtained from

“Binary Data” by D. Collet in *Encyclopedia of Biostatistics* (1998).

Examples

```
data(airway)
summary(airway)
par(mfrow=c(1,2))
plot(age ~ response, data = airway)
plot(duration ~ response, data = airway)
```

`babies` *Crying Babies Data*

Description

The `babies` data frame has 36 rows and 4 columns.

Matched pairs of binary observations concerning the crying of babies. The babies were observed on 18 days and on each day one child was lulled. Interest focuses on the treatment effect “lulling”.

Usage

```
data(babies)
```

Format

This data frame contains the following columns:

`r1` number of children not crying on one day;

`r2` number of children crying on one day;

`lull` indicator variable for the treatment;

`day` factor variable for the days.

Source

The data were obtained from

Cox, D. R. (1970) *Analysis of Binary Data* (page 61). London: Chapman \& Hall.

References

Davison, A. C. (1988) Approximate conditional inference in generalized linear models. *J. R. Statist. Soc.* B, **50**, 445–461.

Examples

```
data(babies)
coplot(r2/(r1+r2) ~ day | lull, data = babies)
##
babies.glm <- glm(formula = cbind(r1, r2) ~ day + lull - 1,
                  family = binomial, data = babies)
babies.cond <- cond(object = babies.glm, offset = lullyes)
babies.cond
##
## If one wishes to avoid the generalized linear model fit:
babies.cond <- cond.glm(formula = cbind(r1, r2) ~ day + lull - 1,
                        family = binomial, data = babies, offset = lullyes)
babies.cond
```

`cond` *Approximate Conditional Inference - Generic Function*

Description

Performs approximate conditional inference.

Usage

```
cond(object, offset, ...)
```

Arguments

`object`	a fitted model object. Families supported are binomial and Poisson with canonical link function (class `glm`), and regression-scale models (class `rsm`).
`offset`	the covariate occurring in the model formula whose coefficient represents the parameter of interest. May be numerical or a two-level factor. In case of a two-level factor, it must be coded by contrasts and not appear as two dummy variables in the model. Can also be a call to a mathematical function (such as `exp`, `sin`, `...`) or to a mathematical operator (`^`, `/`, `...`) applied to a numerical variable. The call must always agree with the label used to identify the corresponding parameter in the fitted model object passed through the `object` argument. Beware that the label includes the identity function `I()` if an arithmetic operator was used. Other function types (e.g. `factor`) and interactions are not admitted.
`...`	absorbs any additional arguments. See `cond.glm` and `cond.rsm` for details.

Details

This function is generic (see `methods`); method functions can be written to handle specific classes of data. Classes which already have methods for this function include: `glm` and `rsm`.

Value

The returned value is an *approximate conditional inference* object. Classes already supported are `cond` and `marg` depending on whether the fitted model object passed through the `object` argument has class `glm` or `rsm`. See `cond.object` or `marg.object` for more details.

References

Brazzale, A. R. (2000) *Practical Small-Sample Parametric Inference*. Ph.D. Thesis N. 2230, Department of Mathematics, Swiss Federal Institute of Technology Lausanne. Chapter 6.

See Also

`cond.glm`, `cond.rsm`, `cond.object`, `marg.object`

Examples

```
## Urine Data
data(urine)
urine.glm <- glm(r ~ gravity + ph + osmo + conduct + urea + log(calc),
                 family = binomial, data = urine)
##
## function call as offset variable
labels(coef(urine.glm))
cond(urine.glm, log(calc))
##
## large estimate of regression coefficient
urine.glm <- glm(r ~ gravity + ph + osmo + conduct + urea + calc,
                 family = binomial, data = urine)
coef(urine.glm)
urine.glm <- glm(r ~ I(gravity * 100) + ph + osmo + conduct + urea + calc,
                 family = binomial, data = urine)
coef(urine.glm)
urine.cond <- cond(urine.glm, I(gravity * 100))
plot(urine.cond, which = 4)

## House Price Data
## Not run:
data(houses)
houses.rsm <- rsm(price ~ ., family = student(5), data = houses)
##
## parameter of interest: scale parameter
houses.marg <- cond(houses.rsm, scale)
plot(houses.marg, which = 2)

## End(Not run)
```

`cond.glm` *Approximate Conditional Inference for Logistic and Loglinear Models*

Description

Performs approximate conditional inference on a scalar parameter of interest in logistic and loglinear models. The output is stored in an object of class cond.

Usage

```
## S3 method for class 'glm'
cond(object, offset, formula = NULL, family = NULL,
         data = sys.frame(sys.parent()), pts = 20,
         n = max(100, 2*pts), tms = 0.6, from = NULL, to = NULL,
         control = glm.control(...), trace = FALSE, ...)
```

Arguments

`object`	a `glm` object. Families supported are binomial and Poisson with canonical link function.
`offset`	the covariate occurring in the model formula whose coefficient represents the parameter of interest. May be numerical or a two-level factor. In case of a two-level factor, it must be coded by contrasts and not appear as two dummy variables in the model. Can also be a call to a mathematical function (such as `exp`, `sin`, `...`) or to a mathematical operator (`^`, `/`, `...`) applied to a numerical variable. The call must always agree with the label used to identify the corresponding parameter in the `glm` object passed through the `object` argument or defined by `formula` and `family`. Beware that the label includes the identity function `I()` if an arithmetic operator was used. Other function types (e.g. `factor`) and interactions are not admitted.
`formula`	a formula expression (only if no `glm` object is defined).
`family`	a `family` object defining the variance function (only if no `glm` object is defined). Families supported are binomial and Poisson with canonical link function.
`data`	an optional data frame in which to interpret the variables occurring in the formula (only if no `glm` object is defined).
`pts`	number of output points (minimum 10) that are calculated exactly. The default is 20.
`n`	approximate number of output points (minimum 50) produced by the spline interpolation. The default is the maximum between 100 and twice `pts`.
`tms`	defines the range MLE +/- `tms` * S.E. where cubic spline interpolation is replaced by polynomial interpolation. The default is 0.6.
`from`	starting value of the sequence that contains the values of the parameter of interest for which output points are calculated exactly. The default is MLE - 3.5 * S.E.
`to`	ending value of the sequence that contains the values of the parameter of interest for which output points are calculated exactly. The default is MLE + 3.5 * S.E.
`control`	a list of iteration and algorithmic constants that controls the GLM fit. See \ `glm.control` for their names and default values.
`trace`	if `TRUE`, iteration numbers will be printed.
`...`	additional arguments, such as `subset` etc., used by the `glm` fitting routine if the `glm` object is defined through `formula` and `family`. See `glm` for their definition and use. The arguments `weights`, `offset` and `contrasts` are not admitted. The returned value is an object of class cond; see `cond.object` for details.

Details

This function is a method for the generic function cond for class `glm`. It can be invoked by calling `cond` for an object of the appropriate class, or directly by calling `cond.glm` regardless of the class of the object. `cond.glm` has also to be used if the `glm` object is not provided throught the `object` argument but specified by `formula` and `family`.

The function `cond.glm` implements several small sample asymptotic methods for approximate conditional inference in logistic and loglinear models. Approximations for both the conditional log likelihood function and conditional tail area probabilities are available (see `cond.object` for details). Attention is restricted to a scalar parameter of interest. The associated covariate can be either numerical or a two-level factor.

Approximate conditional inference is performed by either updating a fitted generalized linear model or defining the model formula and family. All approximations are calculated exactly for `pts` equally spaced points ranging from `from` to `to`. A cubic spline interpolation is used to extend them over the whole interval of interest, except for the range of values defined by MLE +/- `tms` * S.E. where the spline interpolation is replaced by a higher order polynomial interpolation. This is done in order to avoid numerical instabilities which are likely to occur for values of the parameter of interest close to the MLE. Results are stored in an object of class `cond`. Method functions like `print`, `summary` and `plot` can be used to examine the output or represent it graphically. Components can be extracted using `coef`, `formula` and `family`.

Main references for the methods considered are the papers by *Pierce and Peters (1992)* and *Davison (1988)*. More details on the implementation are given in *Brazzale (1999, 2000)*.

Value

The returned value is an object of class `cond`; see `cond.object` for details.

Note

In rare occasions, `cond.glm` dumps because of non-convergence of the function `glm` which is used to refit the model for a fixed value of the parameter of interest. This happens for instance if this value is too extreme. The arguments `from` and `to` may then be used to limit the default range of MLE +/- 3.5 * S.E. A further possibility is to fine-tuning the constants (number of iterations, convergence threshold) that control the GLM fit through the `control` argument.

`cond.glm` may also dump if the estimate of the parameter of interest is large (tipically > 400) in absolute value. This may be avoided by reparametrizing the model.

The output of `cond.glm` may be unreliable if part of the data have a degenerate distribution. For example take the fungal infections treatment data contained in the `fungal` data frame. Of the five 2×2 contingency tables, two (the first and the third) are degenerate. As they make no contribution to the exact conditional likelihood, they should be omitted from the approximate conditional fit.

References

Brazzale, A. R. (1999) Approximate conditional inference for logistic and loglinear models. *J. Comput. Graph. Statist.*, **8**, 1999, 653–661.

Brazzale, A. R. (2000) *Practical Small-Sample Parametric Inference*. Ph.D. Thesis N. 2230, Department of Mathematics, Swiss Federal Institute of Technology Lausanne.

Davison, A. C. (1988) Approximate conditional inference in generalized linear models. *J. R. Statist. Soc.* B, **50**, 445–461.

Pierce, D. A. and Peters, D. (1992) Practical use of higher order asymptotics for multiparameter exponential families (with Discussion). *J. R. Statist. Soc.* B, **54**, 701–737.

See Also

`cond.object`, `summary.cond`, `plot.cond`, `glm`

Examples

```
## Crying Babies Data
data(babies)
babies.glm <- glm(formula = cbind(r1, r2) ~ day + lull - 1,
                  family = binomial, data = babies)
babies.cond <- cond(object = babies.glm, offset = lullyes)
babies.cond
##
## If one wishes to avoid the generalized linear model fit:
babies.cond <- cond.glm(formula = cbind(r1, r2) ~ day + lull - 1,
                        family = binomial, data = babies, offset = lullyes)
babies.cond

## Urine Data
## (function call as offset variable)
data(urine)
urine.glm <- glm(r ~ gravity + ph + osmo + conduct + urea + log(calc),
                 family = binomial, data = urine)
labels(coef(urine.glm))
urine.cond <- cond(urine.glm, log(calc))
##
## (large estimate of regression coefficient)
urine.glm <- glm(r ~ gravity + ph + osmo + conduct + urea + calc,
                 family = binomial, data = urine)
coef(urine.glm)
urine.glm <- glm(r ~ I(gravity * 100) + ph + osmo + conduct + urea + calc,
                 family = binomial, data = urine)
coef(urine.glm)
urine.cond <- cond(urine.glm, I(gravity * 100))

## Fungal Infections Treatment Data (numerical instabilities around the
##                                    MLE)
## (full data analysis)
data(fungal)
fungal.glm <- glm(cbind(success, failure) ~ center + group - 1,
                  family = binomial, data = fungal,
                  control = glm.control(maxit = 50, epsilon = 1e-005))
fungal.cond <- cond(fungal.glm, groupT)
plot(fungal.cond, which = 2)
## (partial data analysis)
fungal.glm <- glm(cbind(success, failure) ~ center + group - 1,
                  family = binomial, data = fungal, subset = -c(1,2,5,6),
                  control = glm.control(maxit = 50, epsilon = 1e-005))
fungal.cond <- cond(fungal.glm, groupT)
plot(fungal.cond, which = 2)
## (Tables 1 and 3 are omitted).
```

---

cond.object *Approximate Conditional Inference Object*

Description

Class of objects returned when performing approximate conditional inference for logistic and log-linear models.

Arguments

Objects of class `cond` are implemented as a list. The following components are included:

	a list whose elements are the spline interpolations of several first order and higher order statistics. They are used to implement the following likelihood quantities: - the profile and modified profile log likelihoods; - the Wald pivots from the unconditional and conditional MLEs; - the profile and modified likelihood roots (the latter one with a suitable continuity correction); - the Lugannani-Rice tail area approximation (with suitable continuity correction); - the correction term used in the higher order statistics; - the information and nuisance parameter aspects. Method functions work mainly on this part of the object. In order to avoid errors in the calculation of confidence intervals and tail probabilities, this part of the object should not be modified.
`workspace` `coefficients`	a 2×2 matrix containing the unconditional and approximate conditional MLEs and their standard errors.
`call`	function call that created the `cond` object.
`formula`	the model formula.
`family`	the variance function.
`offset`	the covariate occurring in the model formula whose coefficient represents the parameter of interest.
`diagnostics`	diagnostics related to the decomposition of the higher order adjustments into an information and a nuisance parameters term. A value larger than 0.2 in absolute value is an index that higher order methods are needed. See *Pierce and Peters (1992)* for details.
`n.approx`	number of output points that have been calculated exactly.
`omitted.val`	range of values omitted in the spline interpolation of some of the higher order statistics considered. The aim is to avoid numerical instabilities around the maximum likelihood estimate.
`is.scalar`	a logical value indicating whether there are any nuisance parameters. If `FALSE` there are none. Main references for the methods considered are the papers by *Pierce and Peters (1992)* and *Davison (1988).* More details on the implementation and the methods considered are given in *Brazzale (1999, 2000).*

Generation

This class of objects is returned from calls to the function `cond.glm`.

Methods

The class cond has methods for `summary`, `plot`, `print`, `coef` and `family`, amongst others.

References

Brazzale, A. R. (1999) Approximate conditional inference for logistic and loglinear models. *J. Comput. Graph. Statist.*, **8**, 653–661.

Brazzale, A. R. (2000) *Practical Small-Sample Parametric Inference*, Ph.D. Thesis N. 2230, Department of Mathematics, Swiss Federal Institute of Technology Lausanne.

Davison, A. C. (1988) Approximate conditional inference in generalized linear models. *J. R. Statist. Soc.* B, **50,** 445–461.

Pierce, D. A. and Peters, D. (1992) Practical use of higher order asymptotics for multiparameter exponential families (with Discussion). *J. R. Statist. Soc.* B, **54**, 701–737.

See Also

`cond.glm`, `summary.cond`, `plot.cond`

`dormicum` *Dormicum Data*

Description

The `dormicum` data frame has 37 rows and 3 columns.

37 children in a pediatric intensive care unit were treated with varying doses and for varying duration with the drug *Dormicum*. The response variable is 1 if withdrawal symptoms were exhibited and 0 otherwise.

Usage

```
data(dormicum)
```

Format

This data frame contains the following columns:

`symp`	indicator of the presence of withdrawal symptoms;
`dose`	the drug dose in mg/kg;
`days`	the number of days treated.

Source

The data were supplied by *Spadille Biostatistik*, Denmark.

References

Mehta, C. R., Patel, N. T. and Senchaudhuri, P. (2000) Efficient Monte Carlo methods for conditional logistic regression. *J. Amer. Statist. Ass.*, **95**, 99–108.

Examples

```
data(dormicum)
par(mfrow = c(1,2))
plot(dose ~ symp, data = dormicum, xlab = "presence of withdrawal symptoms",
     ylab = "treatment dose (mg/kg)")
plot(days ~ symp, data = dormicum, xlab = "presence of withdrawal symptoms",
     ylab = "treatment days")
```

`family.cond` *Use family() on a "cond" object*

Description

This is a method for the function `family()` for objects inheriting from class `cond`. See `family` for the general behaviour of this function.

Usage

```
## S3 method for class 'cond'
family(object, ...)
```

Arguments

`object`	any object from which a `family` object can be extracted.
`...`	absorbs any additional argument.

See Also

`family`

`family.summary.cond` *Use family() on a "summary.cond" object*

Description

This is a method for the function `family()` for objects inheriting from class `summary.cond`. See `family` for the general behaviour of this function.

Usage

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

Arguments

`object`	any object from which a `family` object can be extracted.
`...`	absorbs any additional argument.

See Also

`family`

`fraudulent` *Fraudulent Automobile Insurance Claims Data*

Description

The `fraudulent` data frame has 42 rows and 12 columns.

127 claims arising from automobile accidents in 1989 in Massachusetts (USA). Each claim was classified as either fraudulent or legitimate by consensus among four independent claims adjusters who examined each case file thoroughly. An exploratory analysis by Derrig and Weisberg (1993) identified 10 binary indicators, each of which denotes the presence or absence of a potential fraud characteristic in the claim situation. They fall into three broad groups relating to "Accident" (AC1, AC9 and AC16), "Claimant" (CL7 and CL11), and "Injury" (IJ2, IJ3, IJ4, IJ6 and IJ12).

Usage

```
data(fraudulent)
```

Format

This data frame contains the following columns:

`r1` the number of frauds detected;

`r2` the total number of automobile insurance claims;

`AC1`,`AC9`,`AC16` potential fraud characteristics pertaining to "Accident". The presence of the fraud characteristic is indicated by a 1, the absence is indicated by a 0.

`CL7`,`CL11` potential fraud characteristics pertaining to "Claimer". The presence of the fraud characteristic is indicated by a 1, the absence is indicated by a 0.

`IJ2`,`IJ3`,`IJ4`,`IJ6`,`IJ12` potential fraud characteristics pertaining to "Injury". The presence of the fraud characteristic is indicated by a 1, the absence is indicated by a 0.

Source

The data were supplied by Dr. Richard Derrig of the Automobile Insurers Bureau of Massachusetts.

References

Mehta, C. R., Patel, N. T. and Senchaudhuri, P. (2000) Efficient Monte Carlo methods for conditional logistic regression. *J. Amer. Statist. Ass.*, **95**, 99–108.

Derrig, R. A. and Weisberg, H. I. (1993). Quantitative methods for detecting fraudulent automobile bodily injury claims. *Manuscript.*

Examples

```
data(fraudulent)
summary(fraudulent)
```

---

fungal *Fungal Infections Treatment Data*

Description

The `fungal` data frame has 10 rows and 4 columns.

Clinical trial on the success of a particular treatment for fungal infections. The study was carried out in five different research units. Interest focuses on the treatment effect.

Usage

```
data(fungal)
```

Format

This data frame contains the following columns:

`success` the number of patients that benefited from the treatment;

`failure` the number of patients with no benefit from the treatment;

`group` an indicator variable for treatment (`T`=treatment, `P`=placebo);

`center` a factor variable indicating the research unit where the study was carried out.

Source

The data were supplied by *Sandoz Pharmaceuticals.*

Examples

```
## (full data analysis)
data(fungal)
fungal.glm <- glm(cbind(success, failure) ~ center + group - 1,
                  family = binomial, data = fungal,
                  control = glm.control(maxit = 50, epsilon = 1e-005))
fungal.cond <- cond(fungal.glm, groupT)
plot(fungal.cond, which = 2)
## (partial data analysis)
```

```
fungal.glm <- glm(cbind(success, failure) ~ center + group - 1,
                  family = binomial, data = fungal, subset = -c(1,2,5,6),
                  control = glm.control(maxit = 50, epsilon = 1e-005))
fungal.cond <- cond(fungal.glm, groupT)
plot(fungal.cond, which = 2)
## (Tables 1 and 3 are omitted).
```

plot.cond *Generate Plots for an Approximate Conditional Inference Object*

Description

Creates a set of plots for an object of class cond.

Usage

```
## S3 method for class 'cond'
plot(x = stop("nothing to plot"), from = x.axis[1], to = x.axis[n],
     which = NULL, alpha = 0.05, add.leg = TRUE, loc.leg = FALSE,
     add.labs = TRUE, cex = 0.7, cex.lab = 1, cex.axis = 1,
     cex.main = 1, lwd1 = 1, lwd2 = 2, lty1 = "solid",
     lty2 = "dashed", col1 = "black", col2 = "blue", tck = 0.02,
     las = 1, adj = 0.5, lab = c(15, 15, 5), ...)
```

Arguments

`x`	a cond object. This is assumed to be the result returned by the `cond.glm` function.
`from`	starting value for the x-axis range. The default value has been set by `cond.glm`.
`to`	ending value for the x-axis range. The default value has been set by `cond.glm`.
`which`	which plot should be printed. Admissible values are 2 to 8 corresponding to the choices in the menu below.
`alpha`	the level used to read off confidence intervals; default is 5%.
`add.leg`	if `TRUE`, a legend is added to each plot; default is `TRUE`.
`loc.leg`	if `TRUE`, position of the legend can be located by hand; default is `FALSE`.
`add.labs`	if `TRUE`, labels are added; default is `TRUE`.
`cex, cex.lab, cex.axis, cex.main`	character expansions relative to the standard size of the device to be used for printing text, labels, axes and main title. See `par` for details.
`lwd1, lwd2`	line width used to compare different curves in the same plot; default is `lwd2 = 2` for higher order solutions and `lwd1 = 1` for first order solutions.
`lty1, lty2`	line type used to compare different curves in the same plot; default is `lty2 = "dashed"` for the Wald statistic and `lty1 = "solid"` for the remaining first- and higher order statistics.

`col1, col2`	colors used to compare different curves in the same plot; default is `col2 = "blue"` for higher order solutions, and `col1 = "black"` for the remaining first order statistics.
`tck, las, adj, lab`	further graphical parameters. See `par` for details.
`...`	optional graphical parameters; see `par` for details.

Details

Several plots are produced for an object of class `cond`. A menu lists all the plots that can be produced. They may be one or all of the following ones:

```
 Make a plot selection (or 0 to exit)

1:plot: All
2:plot: Profile and modified profile log likelihoods
3:plot: Profile and modified profile likelihood ratios
4:plot: Profile and modified likelihood roots
5:plot: Modified and continuity corrected likelihood roots
6:plot: Lugannani-Rice approximations
7:plot: Confidence intervals
8:plot: Diagnostics based on INF/NP decomposition

Selection:
```

If no nuisance parameters are presented, a subset of the above pictures is produced. More details on the implementation are given in *Brazzale (1999, 2000).*

This function is a method for the generic function `plot()` for class `cond`. It can be invoked by calling `plot` or directly `plot.cond` for an object of the appropriate class.

Value

A plot is created on the current graphics device.

Side Effects

The current device is cleared. When `add.leg = TRUE`, a legend is added to each plot, and if `loc.leg = TRUE`, it can be set by the user. All screens are closed, but not cleared, on termination of the function.

Note

The diagnostic plots only represent a preliminary version and need further development.

The two panels on the right trace the information and nuisance correction terms, INF and NP, against the likelihood root statistic. These are generally smooth functions and used to approximate the information and nuisance parameter aspects as a function of the parameter of interest, as shown in the two panels on the left. This procedure has the advantage of largely eliminating the numerical instabilities that affect the statistics around the MLE. The circles in the two leftmost panels represent the limit of INF and NP at the MLE calculated exactly using numerical derivatives. All four pictures are intended to give an idea of the order of magnitude of the two correction terms while trying to deal with the numerical problems that likely occur for these kinds of data.

More details can be found in *Brazzale (2000, Appendix B.2).*

References

Brazzale, A. R. (1999) Approximate conditional inference for logistic and loglinear models. *J. Comput. Graph. Statist.*, **8**, 1999, 653–661.

Brazzale, A. R. (2000) *Practical Small-Sample Parametric Inference*, Ph.D. Thesis N. 2230, Department of Mathematics, Swiss Federal Institute of Technology Lausanne.

See Also

`cond.glm`, `cond.object`, `summary.cond`

Examples

```
## Crying Babies Data
data(babies)
babies.glm <- glm(formula = cbind(r1, r2) ~ day + lull - 1,
                  family = binomial, data = babies)
babies.cond <- cond(object = babies.glm, offset = lullyes)
## Not run:
plot(babies.cond)

## End(Not run)

## Urine Data
data(urine)
urine.glm <- glm(r ~ I(gravity * 100) + ph + osmo + conduct + urea + calc,
                 family = binomial, data = urine)
urine.cond <- cond(urine.glm, I(gravity * 100))
plot(urine.cond, which=4)
```

`print.summary.cond` *Use print() on a “summary.cond” object*

Description

This is a method for the function `print()` for objects inheriting from class `summary.cond`. See `print` and `print.default` for the general behaviour of this function and for the interpretation of `digits`.

Usage

```
## S3 method for class 'summary.cond'
print(x, all = x$all, Coef = x$cf, int = x$int, test = x$hyp,
     digits = if(!is.null(x$digits)) x$digits else max(3, getOption("digits")-3),
      ...)
## S3 method for class 'summary.cond'
print(x, all, Coef, int, test, digits, ...)
```

Arguments

`x`	a `summary.cond` object. This is assumed to be the result returned by the `summary.cond` function.
`all`	if `TRUE` all the information stored in the `summary.cond` object is printed, else only a subset of it. The default is `FALSE`.
`Coef`	if `TRUE`, the unconditional and conditional parameter estimates are printed. The default is `TRUE`.
`int`	if `TRUE`, confidence intervals are printed. The default is `TRUE`.
`test`	if `TRUE`, tests statistics and tail probabilities are printed. The default is `FALSE`.
`digits`	number of significant digits to be printed. The default depends on the value of `digits` set by `options`.
`...`	additional arguments.

Details

Changing the default values of `all`, `Coef`, `int` and `test` allows only a subset of the information in the `summary.cond` object to be printed. With `all = FALSE`, one-sided confidence intervals and the Lugannani-Rice tail approximations are omitted. See `summary.cond` for more details.

Note

The amount of information printed may vary depending on whether there are any nuisance parameters.

See Also

`summary.cond`, `cond.object`, `print.default`

Examples

```
## Urine Data
data(urine)
urine.glm <- glm(r ~ gravity + ph + osmo + conduct + urea + calc,
                 family = binomial, data = urine)
urine.cond <- cond(urine.glm, urea)
print(summary(urine.cond, all = TRUE), digits = 4)
print(summary(urine.cond), Coef = FALSE)
```

`rabbits` *Rabbits Data*

Description

The `rabbits` data frame has 10 rows and 4 columns.

Five different doses of penicillin were administered to rabbits suffering from a streptococci infection and the number of recovering rabbits recorded. The rabbits are cross-classified according to whether the drug is administered immediately or delayed by an hour and a half. Interest focuses on whether the delay effects the treatment.

Usage

```
data(rabbits)
```

Format

This data frame contains the following columns:

`cured` the number of rabbits that recovered;

`died` the number of rabbits that died;

`delay` an indicator variable indicating whether the administration of penicillin was delayed by 1 1/2 hours;

`penicil` the penicillin dose.

Source

Unknown.

Examples

```
data(rabbits)
attach(rabbits)
fc <- cured/(cured + died)
coplot(fc ~ log(penicil) | delay, data = rabbits)
```

summary.cond — *Summary Method for Objects of Class "cond"*

Description

Returns a summary list for objects of class cond.

Usage

```
## S3 method for class 'cond'
summary(object, alpha = 0.05, test = NULL, all = FALSE, coef = TRUE,
        int = ifelse( (is.null(test) || all), TRUE, FALSE),
        digits = NULL, ...)
```

Arguments

`object`	a cond object. This is assumed to be the result returned by the `cond.glm` function.
`alpha`	vector of levels for confidence intervals. The default is 5%.
`test`	vector of values of the parameter of interest one wants to test for. If `NULL`, no test is performed. The default is `NULL`.
`all`	logical value; if `TRUE`, all the information stored in the `summary.cond` object is printed, else only a subset of it. The default is `FALSE`.
`coef`	logical value; if `TRUE`, the unconditional and conditional parameter estimates are printed. The default is `TRUE`.
`int`	logical value; if `TRUE` confidence intervals are printed. The default is `TRUE`.
`digits`	number of significant digits to be printed. The default depends on the value of `digits` set by `options`.
`...`	absorbs any additional argument.

Details

This function is a method for the generic function `summary()` for objects of class `cond`. It can be invoked by calling `summary` or directly `summary.cond` for an object of the appropriate class.

Value

A list is returned with the following components.

`coefficients`	a 2×2 matrix containing the unconditional and approximate conditional MLEs and their standard errors.
`conf.int`	a matrix containing, for each level given in `alpha`, the upper and lower confidence bounds derived from several first- and higher order test statistics. One-sided and two-sided confidence intervals are considered. See `cond.object` for details on the test statistics.

signif.tests	a list with two elements. The first (stats) contains, for each value given in test, the values and tail probabilities of several first- and higher order test statistics. See cond.object for details on the test statistics.The second element of the list (qTerm) contains for each tested hypothesis the correction term used in the higher order solutions.
call	the function call that created the cond object.
formula	the model formula.
family	the variance function.
offset	the covariate occurring in the model formula whose coefficient represents the parameter of interest.
alpha	vector of levels used to compute the confidence intervals.
hypotheses	values for the parameter of interest that have been tested for.
diagnostics	information and nuisance parameters aspects; see cond.object for details.
n.approx	number of output points that have been calculated exactly.
all	logical value; if TRUE, all the information stored in the summary.cond object is printed.
cf	logical value; if TRUE, the unconditional and conditional parameter estimates are printed.
int	logical value; if TRUE, confidence intervals are printed.
is.scalar	a logical value indicating whether there are any nuisance parameters. If FALSE there are none.
digits	number of significant digits to be printed.

Note

The amount of information calculated may vary depending on whether there are any nuisance parameters.

See Also

summary, cond.object

Examples

```
## Crying Babies Data
data(babies)
babies.glm <- glm(formula = cbind(r1, r2) ~ day + lull - 1,
                  family = binomial, data = babies)
babies.cond <- cond(object = babies.glm, offset = lullyes)
summary(babies.cond, test = 0, coef = FALSE)
```

`urine` *Urine Data*

Description

The `urine` data frame has 77 rows and 7 columns.

79 urine specimens were analyzed in an effort to determine if certain physical characteristics of the urine might be related to the formation of calcium oxalate crystals.

Usage

```
data(urine)
```

Format

This data frame contains the following columns:

`r` indicator of the presence of calcium oxalate crystals;

`gravity` the specific gravity of the urine, i.e. the density of urine relative to water;

`ph` the pH reading of the urine;

`osmo` the osmolarity of the urine. Osmolarity is proportional to the concentration of molecules in solution (mOsm).

`conduct` The conductivity of the urine. Conductivity is proportional to the concentration of charged ions in solution (mMho milliMho).

`urea` the urea concentration in millimoles per litre;

`calc` the calcium concentration in millimoles per litre.

Source

The data were obtained from

Andrews, D. F. and Herzberg, A. M. (1985) *Data: A Collection of Problems from Many Fields for the Student and Research Worker*, Cambridge: Cambridge University Press.

References

Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap Methods and Their Application* (Example 7.8). Cambridge: Cambridge University Press.

Examples

```
data(urine)
summary(urine)
pairs(urine)
##
data(urine)
urine.glm <- glm(r ~ gravity + ph + osmo + conduct + urea + log(calc),
                 family = binomial, data = urine)
labels(coef(urine.glm))
urine.cond <- cond(urine.glm, log(calc))
##
## (large estimate of regression coefficient)
urine.glm <- glm(r ~ gravity + ph + osmo + conduct + urea + calc,
                 family = binomial, data = urine)
coef(urine.glm)
urine.glm <- glm(r ~ I(gravity * 100) + ph + osmo + conduct + urea + calc,
                 family = binomial, data = urine)
coef(urine.glm)
urine.cond <- cond(urine.glm, I(gravity * 100))
```

Index