# Package: missreg3 (via r-universe)

June 26, 2026

**Type** Package

**Title** Regression Functions for Two-phase Response-selective Sampled Data

**Version** 3.1-1

**Author** Chris Wild <`c.wild@auckland.ac.nz`>, with contributions from Yannan Jiang <`y.jiang@auckland.ac.nz`>.

**Maintainer** Thomas Lumley <`t.lumley@auckland.ac.nz`>

**Description** Performs a variety of regression analyses using semiparametric maximum likelihood for data subject to response selection and two-stage missingness.

**Imports** glmmML,gtools, stats

**License** GPL (>= 2)

**URL** `https://github.com/tslumley/missreg3`

**LazyLoad** yes

**Repository** https://tslumley.r-universe.dev

**Date/Publication** 2026-06-26 01:42:58 UTC

**RemoteUrl** https://github.com/tslumley/missreg3

**RemoteRef** HEAD

**RemoteSha** 399bc6b423bce3dc4ea50406414272cc6d4e2a58

## Contents

---

## `bin2stg` *Binary regression for two-phase sampled data*

---

### Description

Fits binary regression models to data with the two-phase missingness structure. This class includes stratified case-control data.

### Usage

```
bin2stg(formula, weights = NULL, xstrata = NULL,
        obstype.name = "obstype", data, fit = TRUE,
        xs.includes = FALSE, linkname = "logit",
        start = NULL, Qstart = NULL, int.rescale = TRUE,
        off.set = NULL, control = mlefn.control(...),
        control.inner = mlefn.control.inner(...), ...)
```

### Arguments

| | |
|---|---|
| `formula` | A symbolic description of the model to be fitted. If there is only one non-NA level of the response variable presented in the data, that level is treated as "failure" (control). |
| `weights` | An optional vector of weights to be used in the fitting process. Should be `NULL` or a numeric vector. |
| `xstrata` | Specify names of the stratification variables to be used, e.g. `"vname"` or `c("vname1","vname2",...)`. Strata are defined by cross-classification of all levels. |
| `obstype.name` | Name of the variable specifying labels for observations by sampling and variable type: `"uncond"`, `"retro"`, `"xonly"`, `"y\|x"` or `"strata"`. |
| `data` | A data frame containing all the variables required for analysis, including those for `xstrata` and `obstype.name`. |
| `fit` | If `FALSE`, only stratum report will be generated without model fitting.<br>This is useful in providing a data check, or finding internal ordering of the `xstrata` so that `yCuts` can be specified consistently with this ordering. |
| `xs.includes` | `TRUE` if `weights` specified for observations labelled as `"strata"` include those observed at the second phase (i.e. `"retro"` or `"uncond"` observations). |

| | |
|---|---|
| `linkname` | A specification for the model link function. Three choices are provided: `"logit"`, `"probit"` or `"cloglog"`. The default is `"logit"`. |
| `start` | Starting values for the regression parameters. Can be compusory if the program cannot produce a valid starting value at some situations.<br>When only part of the starting parameters are provided, names of these parameters will be used (if specified) to match the design matrix. Zeros will be used as starting values for all other parameters. This is useful when an updated fit is considered. |
| `Qstart` | An optional starting matrix for Pr(Y=i\|Xstratum=j). The first row should be related to the successes (cases) and the second to the failures (controls). Can be compulsory if the program cannot produce a valid starting value at some situations. |
| `int.rescale` | If `TRUE`, all X variables will be standardised first before fitted in the model. |
| `off.set` | Specify an a `priori` known component to be included in the predictors. Should be `NULL` or a numeric vector. |
| `control` | Specify control parameters for the iterations in `mlefn` call. See `mlefn` for details. |
| `control.inner` | Specify control parameters for inner iterations nested within `mlefn` call. See `mlefn` for details. |
| `...` | Further arguments passed to or from related functions. |

## Details

This function fits binary regression models using several links with various types of observations collected at different two-phase sampling schemes. More detailed descriptions of the function and its applications can be found in "Description of the `missreg` Library" (Wild and Jiang).

## Value

| | |
|---|---|
| `missReport` | Matrix containing information on deleted records with missing observations. |
| `StrReport` | Cross tabulation of counts for different levels of `obstype` and Y-values by X-strata. |
| `xStrReport` | Cross tabulation of counts for `obstype` by X-strata when `obstype="xonly"`. |
| `key` | Specify detailed classification for each of the X-strata. |
| `yKey` | Specify the Y variable and its level that the model is constructed for. |
| `fit` | `TRUE` or `FALSE` as its argument. |
| `error` | The error messages returned by `mlefn` call. Non-zero values indicate an unsuccessful fit. |
| `coefficients` | The coefficients matrix with estimates, standard errors, z values and associated p-values. |
| `loglk` | Log-likelihood returned from final `mlefn` call. |
| `score` | Score vector returned from final `mlefn` call. |
| `inf` | Observed information matrix returned from final `mlefn` call. |

| | |
|---|---|
| `fitted` | The fitted values of Y obtained by transforming the linear predictors by the inverse of the link function. |
| `cov` | The asymptotic covariance matrix (inverse of the informnation matrix). |
| `cor` | The asymptotic correlation matrix. |
| `Qmat` | The estimated Pr(Y=i\|Xstratum=j) from the last iteration. |

## Note

The function `summary.bin2stg` provides a complete summary of the regression results including the Wald tests and a regression panel. All related output functions (`print.bin2stg`, `summary.bin2stg` and `print.summary.bin2stg`) don't have help files provided at the moment.

## Author(s)

Chris Wild, Yannan Jiang

## References

Description of the `missreg` Library, Wild and Jiang, 2007.

## Examples

```
data(leprosy1)
leprosy1$age.trans <- 100 * (leprosy1$age + 7.5)^-2
z1 <- bin2stg(leprosy ~ age.trans + scar, data=leprosy1, weights=counts,
              xstrata="age", xs.includes=TRUE)
summary(z1)

data(leprosy2)
leprosy2$age.trans <- 100 * (leprosy2$age + 7.5)^-2
z2 <- bin2stg(cbind(case,control) ~ age.trans + scar, data=leprosy2,
       xstrata="age", xs.includes=TRUE)
summary(z2)

data(leprosy3)
leprosy3$age.trans <- 100 * (leprosy3$age + 7.5)^-2
z3 <- bin2stg(leprosy ~ age.trans + scar, data=leprosy3, weights=counts,
       xs.includes=TRUE)
summary(z3)

data(wilms.sub)
z4 <- bin2stg(cbind(case,control) ~ stage*hist, xstrata=c("stage","inst"),
              xs.includes=TRUE, data=wilms.sub)
summary(z4)

data(trawl)
attach(trawl)
# 265 out of 787 fish in fine net have length over 35  (caught37=NA)
# 353 out of 738 fish in test net have length over 35  (caught37=1)
# So 738 were caught from (estimate) 353*787/265 that entered
#est. pr(caught) assuming all fish over len=35 are caught
```

```
phat <- 738 / (787*353/265)

z5 <- bin2stg(caught37 ~ I(length-35), weights=count, data=trawl,
          start=c(log(phat/(1-phat)),0), Qstart=matrix(c(phat,1-phat)))
summary(z5)

data(lowbirth.bin)
z6 <- bin2stg(sgagp~mumht+bmi+I(bmi^2) + ethnicdb + factor(occ)+ hyper + smoke,
          weights=counts, xstrata=c("ethnicdb","smokedb"),
          obstype.name=c("instudy"), data=lowbirth.bin, xs.includes=FALSE)
summary(z6)
```

---

## `bivbin2stg` *Bivariate binary regression for two-phase sampled data*

### Description

Fits bivariate binary regression models to data with two correlated binary Y-variables and two-phase missingness structure.

### Usage

```
bivbin2stg(formula1, formula2, formula3, weights = NULL,
    xstrata = NULL, obstype.name = "obstype", data,
    fit = TRUE, xs.includes = FALSE, y1samp = TRUE,
    method = "palmgren", start = NULL, Qstart = NULL,
           off.set = NULL, control = mlefn.control(...),
           control.inner = mlefn.control.inner(...), ...)
```

### Arguments

| | |
|---|---|
| `formula1` | A symbolic description of the model to be fitted for Y1, the binary response defining the case-control status of subjects. When the `spml2` method is considered, it provides model formula for Y1\|Y2 where Y2 is another binary response of interest normally observed at the second phase. |
| `formula2` | A symbolic description of the model to be fitted for Y2, the second binary response of interest correlated with Y1. |
| `formula3` | A symbolic description of the model to be fitted quantifying the association between Y1 and Y2. `~1` will fit a constant model. This model is not required when the `spml2` method is considered. |
| `weights` | An optional vector of weights to be used in the fitting process. Should be `NULL` or a numeric vector. |
| `xstrata` | Specify names of the stratification variables to be used, e.g. `"vname"` or `c("vname1","vname2",...)`. Strata are defined by cross-classification of all levels. |
| `obstype.name` | Name of the variables specifying labels for observations by sampling and variable type: `"uncond"`,`"retro"`,`"y|x"`, `"xonly"`, or `"strata"`. |

| | |
|---|---|
| `data` | A data frame containing all the variables required for analysis, including those for `xstrata` and `obstype.name`. |
| `fit` | If `FALSE`, only stratum report will be generated without model fitting. |
| `xs.includes` | `TRUE` if weights specified for observations labelled as `"strata"` include those observed at the second phase (i.e. `"retro"` or `"uncond"` observations). |
| `y1samp` | `TRUE` if Y-strata are defined by the case-control information of Y1 in the population. `FALSE` if Y-strata are defined by both Y1 and Y2 with either "all controls" (Y1=0 and Y2=0) or not (Y1=1 or Y2=1). |
| `method` | Four methods are implemented: `"palmgren"`, `"bahadur"`, `"copula"` and `"spml2"` (see Details for more descriptions). Note that the last method is not available when `y1samp=FALSE`. |
| `start` | Starting values for the regression parameters in Y1-model, Y2-model and the association model when applicable. |
| `Qstart` | An optional starting matrix for Pr(Ystratum=i \| Xstratum=j). Can be compulsory if the program cannot produce a valid starting value at some situations. |
| `off.set` | Specify an a `priori` known component to be included in the predictors. Should be `NULL` or a numeric vector. |
| `control` | Specify control parameters for inner iterations nested within `mlefn` call. See `mlefn` for details. |
| `control.inner` | Specify control parameters for inner iterations nested within `mlefn` call. See `mlefn` for details. |
| `...` | Further arguments passed to or from related functions. |

## Details

This function fits bivariate binary regresison to two correlated binary outcomes Y1 and Y2 using several models with various types observations collected at differnt two-phase sampling schemes.

The joint distribution of Y1 and Y2 given X can be modelled using the marginal distributions of Pr(Y1|X) and Pr(Y2|X) along with an association model between Y1 and Y2. Currently implemented models for this approach include the Palmgren, Bahadur and Copula models. When we are only interested in Pr(Y2|X), another semiparametric approach (called `spml2` method) can be used in terms of a conditional factorisation Pr(Y1|Y2, X)*Pr(Y2|X) both treated parametrically.

More detailed descriptions of this function can be found in "Description of the `missreg` Library" (Wild and Jiang).

## Value

| | |
|---|---|
| `missReport` | Matrix containing information on deleted records with missing observations. |
| `StrReport` | Cross tabulation of counts for different levels of `obstype` and Y-values by X-strata. |
| `xStrReport` | Cross tabulation of counts for `obstype` by X-strata when `obstype="xonly"`. |
| `key` | Specify detailed classification for each of the X-strata. |
| `ykey` | Specify the Y-variables that the model is being constructed for. |

| | |
|---|---|
| `fit` | `TRUE` or `FALSE` as its argument. |
| `error` | The error messages returned by `mlefn` call. Non-zero values indicate an unsuccessful fit. |
| `coefficients` | The coefficients matrix with estimates, standard errors, z values and associated p-values. Will report separately for each marginal model used. |
| `loglk` | Log-likelihood returned from final `mlefn` call. |
| `score` | Score vector returned from final `mlefn` call. |
| `inf` | Observed information matrix returned from final `mlefn` call. |
| `fitted.Y2` | The fitted values of Y2 obtained by transforming the linear predictors by the inverse of the link function. Note that all methods we have implemented evaluate Pr(Y2\|X) which is normally the model of interest. |
| `cov` | The asymptotic covariance matrix (inverse of the information matrix). |
| `cor` | The asymptotic correlation matrix. |
| `Qmat` | The estimated Pr(Ystratum=i\|Xstratum=j) from the last iteration. |
| ... | |

## Note

The function `summary.bivbin2stg` provides a complete summary of the regression results including the Wald tests and a regression model. All related output functions `print.bivbin2stg`, `summary.bivbin2stg` and `print.summary.bivbin2stg` don't have help files provided at the moment.

## Author(s)

Chris Wild, Yannan Jiang

## References

Description of the `missreg` Library, Wild and Jiang, 2007.

## See Also

`bin2stg`

## Examples

```
### SAMPLING ON CASE-CONTROL INFORMATION OF Y1 ONLY ###
data(cotdeath)
z1 <- bivbin2stg(y1~x, y2~x, ~x, weights=wts, data=cotdeath,
                 xs.includes=TRUE, method="palmgren")
summary(z1)

z2 <- bivbin2stg(y1~x, y2~x, ~x, weights=wts, data=cotdeath,
                 xs.includes=TRUE, method="bahadur")
summary(z2)
```

```
z3 <- bivbin2stg(y1~x, y2~x, ~x, weights=wts, data=cotdeath,
                  xs.includes=TRUE, method="copula")
summary(z3)

z4 <- bivbin2stg(y1~x*y2, y2~x, weights=wts, data=cotdeath,
                  xs.includes=TRUE, method="spml2")
summary(z4)

data(infarct)
z5 <- bivbin2stg(sgagp~ethnic+smoked+hyper+mumwt+mumwtc2+agepreg,
  anyinf~smoked+hyper+age1st, ~age1st, weights=count,
  xstrata=c("sex", "gest"), obstype.name="instudy",
  data=infarct, xs.includes=TRUE, method="palmgren")
summary(z5)

### SAMPLING ON ALL CONTROLS (Y1=0 AND Y2=0) OR NOT ###
data(dat00)
z6 <- bivbin2stg(y1~x, y2~x, ~x, weights=wts, data=dat00, y1samp=FALSE,
                 xstrata="v", xs.includes=FALSE, method="palmgren")
summary(z6)
```

---

## `bivlocsc2stg` *Bivariate binary-linear regression for two-phase sampled data*

### Description

Fits bivariate binary-linear regression models to data with two associated response variables, binary Y1 and continuous Y2, and two-phase missingness structure.

### Usage

```
bivlocsc2stg(formula1, formula2, formula3, weights = NULL,
      xstrata = NULL, data, obstype.name = "obstype",
      fit = TRUE, xs.includes = FALSE, off.set = NULL,
      errdistrn = "normal", errmodpars = 6, start = NULL,
      Qstart = NULL, control = mlefn.control(...),
              control.inner = mlefn.control.inner(...), ...)
```

### Arguments

| | |
|---|---|
| `formula1` | A symbolic description of the model to be fitted for Y1\|Y2, where Y1 is the binary response defining the case-control status of subjects and Y2 is a continuous response of interest observed at the second phase. |
| `formula2` | A symbolic description of the location model to be fitted for Y2. |
| `formula3` | A symbolic description of the log-scale model to be fitted for Y2. |
| `weights` | An optional vector of weights to be used in the fitting process. Should be `NULL` or a numeric vector. |

| | |
|---|---|
| `xstrata` | Specify names of the stratification variables to be used, e.g. `"vname"` or `c("vname1","vnames2",...)`. Strata are defined by cross-classification of all levels. |
| `obstype.name` | Name of the variable specifying labels for observations by sampling and variable type: `"uncond"`,`"retro"`,`"y\|x"`, `"xonly"`, or `"strata"`. |
| `data` | The data |
| `fit` | If `FALSE`, only stratum report will be generated without model fitting. |
| `xs.includes` | `TRUE` if weights specified for observations labelled as `"strata"` include those observed at the second phase (i.e. `"uncond"` or `"retro"` observations). |
| `off.set` | Specify an `a priori` known component to be included in the predictors. Should be `NULL` or a numeric vector. |
| `errdistrn` | A specification for the erro distribution. Three choices are provided: standard logistic (`"logistic"`), standard normal (`"normal"`) or student's-t distribution (`"t"`). The default is `"logistic"`. |
| `errmodpars` | Set parameter values for the error distribution. The default is 6 for student's-t distribution. |
| `start` | Starting values for the regression parameters. |
| `Qstart` | An optional starting matrix for Pr(Ystratum=i\|Xstratum=j). |
| `control` | Specify control parameters for the iterations in `mlefn` call. See `mlefn` for details. |
| `control.inner` | Specify control parameters for inner iterations nested within `mlefn` call. See `mlefn` for details. |
| `...` | Further arguments passed to or from related functions. |

## Details

This function extends the application of `SPML2` method when Y2, the second response of interest associated with Y1, is a continuous variable and ideal to be analysed under the location-scale model. In particular, we use a logistic regression model for Y1|Y2 as in `bivbin2stg` when the `SPML2` method is applied, but a linear regression model for Y2 itself. Although the function allows for different error distributions (`"logistic"`, `"normal"`, and `"t"` are implemented so far), only the `normal` is assumed in the strata function and should be used at this stage.

## Value

| | |
|---|---|
| `missReport` | Matrix containing information on deleted records with missing observations. |
| `StrReport` | Cross tabulation of counts for different levels of `obstype` and Y-values by X-strata. |
| `xStrReport` | Cross tabulation of counts for `obstype` by X-strata when `obstype="xonly"`. |
| `key` | Specify detailed classification for each of the X-strata. |
| `ykey` | Vector containing the names of the Y-variables and the names of the level of Ys the model is being constructed for. The sequence is as (name of Y1, name of the level at Y1=1, name of Y2). |
| `fit` | `TRUE` or `FALSE` as its argument. |
| `error` | The error messages returned by `mlefn` call. Non-zero values indicate an unsuccessful fit. |

| | |
|---|---|
| `coefficients` | The coefficients matrix with estimates, standard errors, z-values and associated p-values. |
| `loglk` | Log-likelihood returned from final `mlefn` call. |
| `score` | Score vector returned from final `mlefn` call. |
| `inf` | Observed information matrix returned from final `mlefn` call. |
| `fitted` | The fitted values of Y2 obtained from the model. |
| `cov` | The asymptotic covariance matrix (inverse of the information matrix). |
| `cor` | The asymptotic correlation matrix. |
| `Qmat` | The estimated Pr(Ystratum)=i\|Xstratum=j) from the last iteration. |

## Note

The function `summary.bivlocsc2stg` gives a complete summary of the regression results including the Wald tests and a regression panel. All related output functions (`print.bivlocsc2stg`, `summary.bivlocsc2stg` and `print.summary.bivlocsc2stg`) don't have help files provided at the moment.

## Author(s)

Chris Wild, Yannan Jiang

## References

Description of the `missreg` Library, Wild and Jiang, 2007.

## See Also

`locsc2stg`; `bivbin2stg`

## Examples

```
## Data Generation ##
N <- 5000
x <- rnorm(N)
eps <- rnorm(N)

theta2 <- c(0.5,1,0)
y2 <- theta2[1]+theta2[2]*x+exp(theta2[3])*eps

theta1 <- c(-3,-0.5,1,0.5)
eta1 <- theta1[1]+theta1[2]*y2+theta1[3]*x+theta1[4]*y2*x
p1 <- plogis(eta1)
y1 <- 1*(runif(N)<p1)

xcut <- c(-30,-1,0,1,30)
xstrata <- as.numeric(cut(x,xcut))

indca <- (1:N)[y1==1]
indct <- sample((1:N)[y1==0],length(indca))
```

```
ind <- sort(c(indca,indct))
rest <- (1:N)[-ind]
obstype <- rep("retro",N)
obstype[rest] <- "strata"
y2[rest] <- NA; x[rest] <- NA
dat <- data.frame(y1,y2,x,xstrata,obstype)

## Proportion of cases in population (about 0.1) ##
prca <- length(indca)/N
prca

## Model fit ##
z <- bivlocsc2stg(y1~y2*x,y2~x,~1,xstrata="xstrata",data=dat,xs.includes=FALSE)
summary(z)
```

---

## `leprosy1` *The Leprosy data*

### Description

The leprosy data set was described in Scott and Wild (1997, 2001) and used as an example of standard two-phase case-control sampled data.

### Usage

```
data(leprosy1)
```

### Format

A data frame with 42 observations on the following 5 variables.

`leprosy` a factor with levels `no` `yes`

`age` a numeric vector indicating the mid-point of six 5-year age groups

`scar` a factor with levels `no` `yes`

`counts` a numeric vector indicating number of subjects with each observation

`obstype` a factor with levels `retro` and `strata`, compulsory for function call

### Details

The leprosy data were obtained by sampling from the results of a population cross-sectional study of people under 35 in Northern Malawi and represented as a three-way contingency table in Clayton and Hills (1993). Those people with leprosy were defined to be cases and the rest to be controls. The data were first categorized into six 5-year age sampling strata and the numbers of cases and controls falling into each stratum were observed. All cases have been chosen with an equal-sized control group subsampled from the control population within each age stratum. The potential risk factor that indicates the presence or absence of a BCG vaccination scar was then observed.

The data are represented in three different formats in `leprosy1`, `leprosy2` and `leprosy3`. See "Description of the `missreg` Library" for more details.

## References

Description of the `missreg` Library, Wild and Jiang, 2007

## Examples

```
data(leprosy1)
```

---

| `linbin2stg` | *Estimate binary-logistic parameters and odds ratios using linear regression with a single continuous Y-variable for two-phase sampled data.* |
|---|---|

## Description

Fit location-scale model of the form `Y = eta + sigma*error` to data with a single continuous Y-variable and two-phase missingness structure, and convert to binary-logistic parameters and odds-ratio estimates with appropriate cut-points of Y.

## Usage

```
linbin2stg(formula1, yCuts, lower.tail = TRUE, weights = NULL,
 xstrata = NULL, data = list(), obstype.name = "obstype",
 fit = TRUE, xs.includes = FALSE, compactX = FALSE,
 start = NULL, Qstart = NULL, deltastart = NULL,
 int.rescale = TRUE, control = mlefn.control(...),
 control.inner = mlefn.control.inner(...), ...)
```

## Arguments

| | |
|---|---|
| `formula1` | A symbolic description of the location model to be fitted, i.e. eta. |
| `yCuts` | Cutpoint(s) used to define the binary Y-variable for logistic regression. Can be a matrix form (1*S) with S the number of xstrata. |
| `lower.tail` | If TRUE, define the cases being `{Y <= yCuts}`. |
| `weights` | An optional vector of weights to be used in the fitting process. Should be `NULL` or a numeric vector. |
| `xstrata` | Specify names of the stratification variables to be used, e.g. `"vname"` or `c("vname1","vname2",...)`. Strata are defined by cross-classification of all levels. |
| `data` | A data frame containing all the variables required for analysis, including those for `xstrata` and `obstype.name`. |
| `obstype.name` | Name of the variable specifying labels for observations by sampling and variable type: `"uncond"`, `"retro"`, `"xonly"`, `"y\|x"` or `"strata"`. |
| `fit` | If `FALSE`, only stratum report will be generated without model fitting. |
| `xs.includes` | `TRUE` if weights specified for observations labelled as `"strata"` include those observed at the second phase (i.e. `"retro"` or `"uncond"` observations). |

| | |
|---|---|
| `compactX` | If `TRUE`, compress X matrix to distinct values with counts before model fitting. |
| `start` | Starting values for the regression parameters. Can be compusory if the program cannot produce a valid starting value at some situations. |
| `Qstart` | An optional starting matrix for Pr(Ystratum=i\|Xstratum=j). Can be compulsory if the program cannot produce a valid starting value at some situations. |
| `deltastart` | An optional starting matrix for Pr(X=xk\|Xstratum=j). |
| `int.rescale` | If `TRUE`, all X-variables will be standardised first before fitted in the model. |
| `control` | Specify control parameters for the iterations in `mlefn` call. See `mlefn` for details. |
| `control.inner` | Specify control parameters for inner iterations nested within `mlefn` call. See `mlefn` for details. |
| `...` | Further arguments passed to or from related functions. |

## Details

This function is a simple application of `locsc2stg` fitting linear regression models with a continuous Y using logistic error distribution. The results are then converted to much more efficient inferences about the same odds-ratio parameters being estimated by the logistic regression with the dichotomized binary outcome (case-control).

More detailed descriptions of this function can be found in "Description of the `missreg` Library" (Wild and Jiang).

## Value

| | |
|---|---|
| `missReport` | Matrix containing information on deleted records with missing observations. |
| `StrReport` | Cross tabulation of counts for different levels of `obstype` and Y-values by X-strata. |
| `xStrReport` | Cross tabulation of counts for `obstype` by X-strata when `obstype="xonly"`. |
| `key` | Specify detailed classification for each of the X-strata. |
| `yCutsKey` | Specify the cutoff intervals for defined Y-strata within each X-stratum. |
| `fit` | `TRUE` or `FALSE` as its argument. |
| `error` | The error messages returned by `mlefn` call. Non-zero values indicate an unsuccessful fit. |
| `coefficients` | Linear regression coefficients. |
| `loglk` | Log-likelihood returned from final `mlefn` call. |
| `score` | Score vector returned from final `mlefn` call. |
| `inf` | Observed information matrix returned from final `mlefn` call. |
| `fitted` | The fitted values of Y obtained from the model. |
| `cov` | The asymptotic covariance matrix (inverse of the informnation matrix) of linear parameter estimates. |
| `cor` | The asymptotic correlation matrix of linear parameter estimates. |
| `bcoefficients` | Binary regression coefficients converted from linear parameters. |
| `bcov` | The asymptotic variance of binary parameter estimates. |

## Note

The function `summary.linbin2stg` provides a complete summary of the regression results including the Wald tests and a regression panel for linear coefficients, a regression panel for binary coefficients, and associated odds-ratio estimates and confidence intervals. All related output functions (`print.linbin2stg`, `summary.linbin2stg` and `print.summary.linbin2stg`) don't have help files provided at the moment.

Also note that the intercept of binary coefficients will not be available when more than one cut-point of Y is used, e.g. different for each x-stratum.

## Author(s)

Chris Wild, Yannan Jiang

## References

Description of the `missreg` Library, Wild and Jiang, 2007.

## See Also

`locsc2stg`

## Examples

```
data(lowbirth.ls)
lowbirth.ls$sex.age <- interaction(lowbirth.ls$sex,lowbirth.ls$gest)
yCuts <- matrix(c(2550,2650,2740,2840,2900,3010,3030,3140),nrow=1)
yCut1 <- mean(yCuts)

### Multiple yCuts;
z1 <- linbin2stg(birthwt~gest+mumht+bmi+ethnicdb+hyper+smoke,
                 yCuts=yCuts, xstrata=c("sex.age"), data=lowbirth.ls,
                 obstype.name=c("instudy"), xs.includes=FALSE)

summary(z1)

### Single yCut;
z2 <- linbin2stg(birthwt~gest+mumht+bmi+ethnicdb+hyper+smoke,
                 yCuts=yCut1, xstrata=c("sex.age"), data=lowbirth.ls,
                 obstype.name=c("instudy"), xs.includes=FALSE)

summary(z2)
```

---

# `locsc2stg` *Linear regression with location-scale model for two-phase sampled data.*

## Description

Fits location-scale model of the form `Y = eta1 + exp(eta2)*error` to data with a single continuous Y-variable and two-phase missingness structure, using the linear predictors `eta1` and `eta2` for specification of the location and scale respectively.

## Usage

```
locsc2stg(formula1, formula2, yCuts=NULL, weights=NULL,
   xstrata=NULL, data=list(), obstype.name="obstype",
   method="direct", fit=TRUE, errdistrn="logistic",
   errmodpars=6, xs.includes=FALSE, compactX=FALSE,
   compactY=TRUE, straty.maxnvals=20, start=NULL,
   Qstart=NULL, deltastart=NULL, int.rescale=TRUE,
          control=mlefn.control(...),
          control.inner=mlefn.control.inner(...), ...)
```

## Arguments

| | |
|---|---|
| `formula1` | A symbolic description of the location model to be fitted (eta1). |
| `formula2` | A symbolic description of the log-scale model to be fitted (eta2). `~1` will fit a constant. |
| `yCuts` | Cutpoints used to define Y-strata. Critical when `method="ycutmeth"`. Also required when `method="direct"` but the starting values are not provided (See Details for more descriptions). |
| `weights` | An optional vector of weights to be used in the fitting process. Should be `NULL` or a numeric vector. |
| `xstrata` | Specify names of the stratification variables to be used, e.g. `"vname"` or `c("vname1","vname2",...)`. Strata are defined by cross-classification of all levels. |
| `obstype.name` | Name of the variable specifying labels for observations by sampling and variable type: `"uncond"`, `"retro"`, `"xonly"`, `"y\|x"` or `"strata"`. |
| `data` | A data frame containing all the variables required for analysis, including those for `xstrata` and `obstype.name`. |
| `method` | Two methods are implemented: `"ycutmeth"` and `"direct"` (see Details for more descriptions). |
| `fit` | If `FALSE`, only stratum report will be generated without model fitting. |
| `errdistrn` | A specification for the error distribution. Three choices are provided: standard logistic (`"logistic"`), standard normal (`"normal"`) or student's-t distribution (`"t"`). The default is `"logistic"`. |
| `errmodpars` | Set parameter values for the error distribution. The default is 6 for student's-t distribution. |
| `xs.includes` | `TRUE` if weights specified for observations labelled as `"strata"` include those observed at the second phase (i.e. `"retro"` or `"uncond"` observations). |
| `compactX` | If `TRUE`, compress X matrix to distinct values with counts before model fitting. This is only applicable to `method="direct"`. |

| | |
|---|---|
| `compactY` | If TRUE, limit the Y-values observed at the first phase (`obstype="strata"`) to limited numbers of equally spaced possible values. This is only applicable to `method="direct"`. |
| `straty.maxnvals` | If `compactY=TRUE`, specify the number of equally spaced possible values spanning the range of Y observed as `"strata"`. The default is 20. |
| `start` | Starting values for the regression parameters. Can be compusory if the program cannot produce a valid starting value at some situations. |
| `Qstart` | An optional starting matrix for Pr(Ystratum=i\|Xstratum=j). Can be compulsory if the program cannot produce a valid starting value at some situations. |
| `deltastart` | An optional starting matrix for Pr(X=xk\|Xstratum=j). This is only applicable to `method="direct"`. |
| `int.rescale` | If `TRUE`, all X-variables will be standardised first before fitted in the model. |
| `control` | Specify control parameters for the iterations in `mlefn` call. See `mlefn` for details. |
| `control.inner` | Specify control parameters for inner iterations nested within `mlefn` call. See `mlefn` for details. |
| `...` | Further arguments passed to or from related functions. |

## Details

This function fits location-scale models to continuous Y using different error distributions with various types of observations collected at different two-phase sampling schemes. More detailed descriptions of this function can be found in "Description of the `missreg` Library" (Wild and Jiang).

Two methods are implemented with either Y being categorical (`"ycutmeth"`) or at a continuous scale (`"direct"`). The argument `yCuts` is critical to the first approach but only required for the second approach when a starting value is needed. If `yCuts` is a vector, it defines the Y-strata with intervals (-infty, `yCuts`, infty). If `yCuts` is a matrix, the number of columns indicates the number of strata used and you can define different cutpoints for each stratum. If you want to have differing numbers of cutpoints for different X-strata, pad out the bottom of any column that is not full with NAs.

## Value

| | |
|---|---|
| `missReport` | Matrix containing information on deleted records with missing observations. |
| `StrReport` | Cross tabulation of counts for different levels of `obstype` and Y-values by X-strata. |
| `xStrReport` | Cross tabulation of counts for `obstype` by X-strata when `obstype="xonly"`. |
| `key` | Specify detailed classification for each of the X-strata. |
| `yCutsKey` | Specify the cutoff intervals for defined Y-strata within each X-stratum. |
| `fit` | `TRUE` or `FALSE` as its argument. |
| `error` | The error messages returned by `mlefn` call. Non-zero values indicate an unsuccessful fit. |
| `coefficients` | The coefficients matrix with estimates, standard errors, z values and associated p-values. |

| | |
|---|---|
| `loglk` | Log-likelihood returned from final `mlefn` call. |
| `score` | Score vector returned from final `mlefn` call. |
| `inf` | Observed information matrix returned from final `mlefn` call. |
| `fitted` | The fitted values of Y obtained from the model. |
| `cov` | The asymptotic covariance matrix (inverse of the informnation matrix). |
| `cor` | The asymptotic correlation matrix. |
| `Qmat` | The estimated Pr(Ystratum=i\|Xstratum=j) from the last iteration. |
| `deltamat` | The estimated `delta` matrix from the last iteration.<br>This is only applicable to `method="direct"`. |

## Note

The function `summary.locsc2stg` provides a complete summary of the regression results including the Wald tests and a regression panel. All related output functions (`print.locsc2stg`, `summary.locsc2stg` and `print.summary.locsc2stg`) don't have help files provided at the moment.

## Author(s)

Chris Wild, Yannan Jiang

## References

Description of the `missreg` Library, Wild and Jiang, 2007.

## See Also

`bin2stg`

## Examples

```
data(lowbirth.ls)
lowbirth.ls$sex.age <- interaction(lowbirth.ls$sex,lowbirth.ls$gest)
yCuts <- matrix(c(2550,2650,2740,2840,2900,3010,3030,3140),nrow=1)

z1 <- locsc2stg(birthwt ~ gest + mumht + bmi+ethnicdb+hyper+smoke, ~1,
  yCuts=yCuts, xstrata=c("sex.age"), data=lowbirth.ls,
   obstype.name=c("instudy"), xs.includes=FALSE,
  method="ycutmeth")
summary(z1)

z2 <- locsc2stg(birthwt ~ gest + mumht + bmi+ethnicdb+hyper+smoke,~1,
                xstrata=c("sex.age"),data=lowbirth.ls,
         obstype.name=c("instudy"), xs.includes=FALSE,
  method="direct", start=z1$coefficients, compactX=TRUE,
                compactY=TRUE, straty.maxnvals=20)
summary(z2)

z2 <- locsc2stg(birthwt ~ gest + mumht + bmi+ethnicdb+hyper+smoke,~1,
  yCuts=yCuts, xstrata=c("sex.age"), data=lowbirth.ls,
```

```
  obstype.name=c("instudy"), xs.includes=FALSE,
  method="direct", start=z1$coefficients, Qstart=z1$Qmat,
  compactX=TRUE, compactY=TRUE, straty.maxnvals=20,
         control.inner=mlefn.control.inner(n.earlyit=3))
summary(z2)

z3 <- locsc2stg(birthwt ~ gest + mumht + bmi+ethnicdb+hyper+smoke,~1,
                xstrata=c("sex.age"),data=lowbirth.ls,
  obstype.name=c("instudy"), xs.includes=FALSE,
  method="direct", start=z2$coefficients,
  deltastart=z2$deltamat, compactX=TRUE,
           compactY=TRUE, straty.maxnvals=100)
summary(z3)
```

---

## `lowbirth.bin` *The Low Birthweight data*

### Description

A subset of the data collected in the Auckland Birthweight Collaborative (ABC) Study.

### Usage

```
data(lowbirth.bin)
```

### Format

A data frame with 1148 observations on the following 18 variables.

**sgagp** A factor defining the case (sga) and control (aga) status of the baby.

**sex** 1=female or 2=male

**instudy** A factor with levels `retro` and `strata`, used as the `obstype` variable in function call

**htstrat** A factor with levels as class intervals of mother's height

**ethnicdb** A factor with levels `A` (Asian), `E` (Euro.), `M` (Maori) or `P` (Pacifican)

**mstrat** Marital status of the mother

**occ** Mother's ccupational group, 1 to 3 (3 is highest)

**mumht** Height of the mother in cm

**mumwt** Weight of the mother in kg

**bmi** Body mass index of the mother

**smoke** Smoking status of the mother prior to pregnancy

**smokedb** Smoking variable from database

**age1st** Mother's age at first pregnancy

**hyper** Any hypertension (1=yes, 0=no)

**edstratdb** Mother's educational level

**eductrm** Mother's age left school

**mstratdb** As for `mstrat` with some levels combined

**counts** Number of subjects with each observation (frequency)

### Details

The ABC study was conducted in 1995-1997 in order to find potential risk factors associated with small-for-gestational-age babies in New Zealand. It was a population-based case-control study with the cases being those babies with their birthweights equal to or below the sex-specific 10th percentile for gestational age in the population.

The `lowbirh.bin` is a semi-random subset of the original data.

### References

Description of the `missreg` Library, Wild and Jiang, 2007.

---

## `MEtaProspModInf` *Prospective Model Information function for models with M linear predictors.*

### Description

A sub-function called by `ML2Inf` to supply values and its derivatives for the first part of the profile loglikelihood regarding to the model of interest using the discrete partition version.

### Usage

```
MEtaProspModInf(theta,nderivs=2,y,x,wts=1,modelfn,off.set=0, ...)
```

### Arguments

| | |
|---|---|
| `theta` | Vector of the parameter values. |
| `nderivs` | Number of derivatives to be calculated, ranged from 0 (loglikelihood only) to 2 (information matrix). |
| `y` | The response of interest, can be either a vector or matrix. |
| `x` | A 3-dimensional array (R*C*M) specifying the covariates values, with R the number of observations, C the length of `theta` and M the number of linear predictors used. |
| `wts` | An optional vector of weights (`n_i`) to be used in the fitting process. The default is 1. |
| `modelfn` | A class of sub-functions called by `MEtaProspModInf` to calculate the values and their derivatives with respect to the linear predictor (`eta`'s) of X for the model of interest `f(Y\|X; theta)`. |
| `off.set` | The offset provided in a matrix form (R*M) with R the number of observations and M the number of linear predictors used. |
| `...` | Further arguments passed to or from related functions. |

### Details

This sub-function is used to implement prospective regression models with a fixed number of M linear predictors. It calculates the value and its derivatives for the first part of the profile loglikelihood in the form of `l*(theta,Q)` within each s-stratum

`sum_{A(s)}{n_i^(s)*log{f(y_i^{(s)}|x_i^{(s)};theta)}}`,

with respect to `theta` through the M linear predictors (m=1,...,M),

`eta_{im}` = `o_{im}+x_{i(m)}^T*theta`

See "Description of the `missreg` Library" for all details.

### Value

A list with the following components

| | |
|---|---|
| `loglk` | Log-likelihood obtained from the current `theta` values |
| `score` | Score vector obtained from the curent `theta` values when `nderivs>=1`; `NULL` otherwise. |
| `inf` | Observed information matrix obtained from the current `theta` values when `nderivs=2`; `NULL` otherwise. |

### Author(s)

Chris Wild, Yannan Jiang

### References

Description of the `missreg` Library, Wild and Jiang, 2007.

### See Also

`ML2Inf`

---

## `ML2directInf` *Core likelihood calculation function for the direct approach*

### Description

A sub-function called by `MLdirectInf` to provide the value, score vector and information matrix at `theta` for the so-called profile loglikelihood `l_P(theta)` of the form `l(theta, delta)` within each s-stratum with stratified two-phase sampled data. It reduces to an unstratified approach when nstrata=1.

## Usage

```
ML2directInf(theta, nderivs = 2, modelfn, hmodelfn, x, y, Aposn,
      Acounts, Bposn, Bcounts, hvalue, Cmult, delta,
      off.set = matrix(0, dim(x)[1], dim(x)[3]), inxStrat,
      control.inner = mlefn.control.inner(...), ...)
```

## Arguments

| | |
|---|---|
| `theta` | Starting values for parameters `theta` in the regression model. |
| `nderivs` | Number of derivatives to be calculated. Either 0 (loglikelihood value only), 1 (also return score vector), or 2 (also return information matrix). |
| `modelfn` | A class of sub-functions called by `ML2directInf` to supply values and their derivatives with respect to the `eta`'s (M linear predictors with respect to `theta`) for the model of interest `f(Y\|X;theta)`. |
| `hmodelfn` | A class of sub-functions called by `ML2directInf` to supply values and their derivatives with respect to the `eta`'s for `pr(h_k\|x_j;theta)` under the same class of models. |
| `x` | A 3-dimensional array (R*C*M) specifying the covariates values, with R the number of observations, C the length of `theta` and M the number of linear predictors used. |
| `y` | The response of interest, can be eitehr a vector or matrix. |
| `Aposn` | A vector specifying the positions of those observations with the set of complete (x, y)-values from s-stratum. |
| `Acounts` | A vector specifying the frequency of each observation (`n_i`) with the set of complete (x,y)-values from s-stratum. |
| `Bposn` | A vector specifying the positions of those observations with the x-values observed in s-stratum; `NULL` in prospective sampling. |
| `Bcounts` | A vector specifying the frequency of each observation (`m_j`) with the x-values observed in s-stratum; `NULL` in prospective sampling. |
| `hvalue` | The `h_k^(s)` in the loglikelihood. |
| `Cmult` | The `r_k^(s)` in the loglikelihood. |
| `delta` | The `delta_j^(s)` in the loglikelihood. |
| `off.set` | The offset provided in a matrix form (R*M) with R the number of observations and M the number of linear predictors used. |
| `inxStrat` | See `ML2Inf`. |
| `control.inner` | Specify control parameters for inner iterations nested within the `mlefn` function call. See `mlefn` for details. |
| `...` | Further arguments passed to or from related functions. |

## Details

This is the core function in the direct approach to calculate the value, score vector and observed information matrix at `theta` for the profile loglikelihood `l_P(theta)` of the form `l^(s)(theta,delta^(s))` within each s-stratum.

It is an inner function called by `MLdirectInf`.

## Value

A list with the following components.

| | |
|---|---|
| `loglk` | Log-likelihood value obtained from the current `theta` values. |
| `score` | Score vector obtained from the current `theta` values when `nderivs>=1`; `NULL` otherwise. |
| `inf` | Observed information matrix obtained from the current `theta` values when `nderivs=2`; `NULL` otherwise. |
| `delta` | A vector of length J providing values for `delta_j^(s)` either as its entry values or updated from the inner iterative process when `Bposn` is not `NULL`. |

## Author(s)

Chris Wild, Yannan Jiang

## References

Description of the `missreg` Library, Wild and Jiang, 2007.

## See Also

`MLdirectInf`; `ML2Inf`

---

`ML2Inf` *Core likelihood calculation function for the discrete partition version*

## Description

A sub-function called by `MLInf` to provide the value, score vector and information matrix at `theta` for the so-called profile loglikelihood `l*(theta, Q)` within each stratum with stratified two-phase sampled data. It reduces to an unstratified approach when nstrata=1.

## Usage

```
ML2Inf(theta, nderivs = 2, ProspModInf, StratModInf, x, y, Aposn,
       Acounts, Bposn, Bcounts, rvec, Qs, usage = "thetaonly",
       thetaparts = 0, paruse = "auto", inxStrat,
       off.set = matrix(0, dim(x)[1], dim(x)[3]),
       control.inner = mlefn.control.inner(...), ...)
```

## Arguments

| | |
|---|---|
| `theta` | Starting values for parameters `theta` in the regression model unless `usage` below is `"combined"`, in which case `theta` should contain starting values for the `theta` parameters followed by starting values fo the `rho` or `xi` parameters (which will be stripped off on entry to the function). |
| `nderivs` | Number of derivatives to be calculated. Either 0 (loglikelihood value only), 1 (also return score vector), or 2 (also return information matrix). |
| `ProspModInf` | A class of sub-functions called by `ML2Inf` to supply values and their derivatives for the first (A) part of the loglikelihood regarding to the model of interest. |
| `StratModInf` | A class of sub-functions called by `ML2Inf` to supply values and their derivatives for the third (B) part of the loglikelihood regarding to the `h`-distribution at the first phase. |
| `x` | See `MEtaProspModInf`. |
| `y` | See `MEtaProspModInf`. |
| `Aposn` | A vector specifying the positions of those observations contributed to the A part of the loglikelihood in the data matrix. |
| `Acounts` | A vector specifying the frequency of each observation (`n_i`) contributed to the A part of the loglikelihood. |
| `Bposn` | A vector specifying the positions of those observations contributed to the B part of the loglikelihood in the data matrix. |
| `Bcounts` | A vector specifying the frequency of each observation (`m_j`) contributed to the B part of the loglikelihood. |
| `rvec` | The `r_k^(s)` in the loglikelihood. |
| `Qs` | The `Q_k^(s)` in the loglikelihood. |
| `usage` | Work with and report results for the following three conditions: (1) `"thetaonly"` (profile other parameters); (2) `"combined"` (fit both `theta` and `rho/xi` simultaneously); and (3) `"Qfixed"` (fix other parameters). |
| `thetaparts` | A vector of length 2 specifying the number of `theta` and `rho/xi` parameters as appropriate; Used only if `usage="combined"`. |
| `paruse` | The choice of using either `rho` or `xi` parameters as follows: (1) `"rhos"`; (2) `"xis"`; or (3) `"auto"` (function to choose the `rho's} if` no more than one `\code{r_k^(s)=0} and the \code{xi's` otherwise). Any other string is treated as the last option. |
| `inxStrat` | Optional to enable printing a diagnostic when `ML2Inf` fails and has been called from `MLInf`. |
| `off.set` | See `MEtaProspModInf`. |
| `control.inner` | Specify control parameters for inner iterations nested within the `mlefn` function call. |
| `...` | Further arguments passed to or from related functions. |

## Details

This is the core function at the distrete partition version to calculate the value, score vector and observed information matrix at `theta` for the so-called profile loglikelihood `l*^(s)(theta,Q^(s))`. It is an inner function called by `MLInf` to supply values of `l*` and its derivatives within each s-stratum.

See "Description of the `missreg` Library" for more details.

## Value

A list with the following components.

| | |
|---|---|
| `loglk` | Log-likelihood obtained from the current `theta` values. |
| `score` | Score vector obtained from the current `theta` values when `nderivs>=1`; `NULL` otherwise. |
| `inf` | Observed information matrix obainted from the current `theta` values when `nderivs=2`; `NULL` otherwise. |
| `Qs` | A vector of length K providing values for `Q_k^(s)` either as its entry values or updated from the inner iterative process when `r_k^(s)` is not equal to 0 and `usage="thetaonly"`. |

## Author(s)

Chris Wild, Yannan Jiang

## References

Description of the `missreg` Library, Wild and Jiang, 2007.

## See Also

`MLInf`; `MEtaProspModInf`

---

# `MLdirectInf` *Likelihood calculation function for the direct approach with stratification*

## Description

An outer function of `ML2directInf` to provide the value, score vector and information matrix at `theta` for the profile loglikelihood `l_P(theta)` of the form `l(theta,delta)` with stratified two-phase sampled data. It reduces to an unstratified approach when nstrata=1.

## Usage

```
MLdirectInf(theta, nderivs = 2, deltamat = NULL, modelfn,
     hmodelfn, x, y, xStrat, Aposn, Acounts, Bposn,
     Bcounts, hvalue, Cmult, hxStrat,
     off.set = matrix(0, dim(x)[1], dim(x)[3]), extra = NULL,
     control.inner = mlefn.control.inner(...), ...)
```

## Arguments

| | |
|---|---|
| `theta` | Starting values for parameter `theta` in the regression model. |
| `nderivs` | Number of derivatives to be calculated. See `ML2directInf` for details. |
| `deltamat` | The `delta_j^(s)` provided in a matrix form (J*S) with S the number of strata and J the number of distinct x-values observed. |
| `modelfn` | See `ML2directInf`. |
| `hmodelfn` | See `ML2directInf`. |
| `x` | See `ML2directInf`. |
| `y` | See `ML2directInf`. |
| `xStrat` | A vector of values 1 to S specifying the stratum membership of each observation. |
| `Aposn` | A vector specifying the positions of those observations with the set of complete (x,y)-values. |
| `Acounts` | A vector specifying the frequency of each observation (`n_i`) with the set of complete (x,y)-values. |
| `Bposn` | A vector specifying the positions of those observations with the x-values observed; `NULL` in prospective sampling. |
| `Bcounts` | A vector specifying the frequency of each observation (`m_j`) with the x-values observed; `NULL` in prospective sampling. |
| `hvalue` | The `h_k` in the loglikelihood. |
| `Cmult` | The `r_k` in the loglikelihood. |
| `hxStrat` | A vector of value 1 to S specifying the stratum membership of each `hvalue`. |
| `off.set` | See `ML2directInf`. |
| `extra` | Provides `deltamat` from last iteration as starting values for next inner iterative loop in `mlefn` function call. |
| `control.inner` | See `ML2directInf`. |
| `...` | Further arguments passed to or from related functions. |

## Details

This is the direct function called by `mlefn` to calculate the value, score vector and observed information matrix at `theta` for the so-called profile loglikelihood `l_P(theta)` using the direct approach. It calls the inner function `ML2directInf` to evaluate `l^(s)(theta,delta^(s))` within each s-stratum.

## Value

A list with the following components.

| | |
|---|---|
| `loglk` | Log-likelihood obtained from the current `theta` values. |
| `score` | Score vector obtained from the current `theta` values when `nderivs>=1`; `NULL` otherwise. |
| `inf` | Observed information matrix obtained from the current `theta` values when `nderivs=2`; `NULL` otherwise. |
| `extra` | A list providing updated `deltamat` values from previous iteration. |

## Author(s)

Chris Wild, Yannan Jiang

## References

Description of the `missreg` Library, Wild and Jiang, 2007.

## See Also

`ML2directInf`; `mlefn`

---

# `mlefn` *A modified Newton maximiser*

## Description

A function to maximise, minimise or find stationary values for a (general) function. It was originally written to maximize a loglikelihood function so that is the language that is employed.

`mlefn.control` and `mlefn.control.inner` supply parameter values to control the iterative process and reporting. They differ only in their defaults.

## Usage

```
mlefn(theta, loglkfn, control=mlefn.control(...), ...)

mlefn.control(messg="", niter=20, tol=1e-08, guide="uphill",
      print.progress=2, max.eigenrat=0.05, n.earlyit=0,
      constrain="no", fixed=NULL, Aconstrain=NULL,
      cconstrain=NULL, ...)

mlefn.control.inner(messg="Inner:", niter=20, tol=1e-08,
      guide="auto", print.progress=0, max.eigenrat=0.05,
      n.earlyit=0, constrain="no", fixed=NULL,
      Aconstrain=NULL, cconstrain=NULL, ...)
```

## Arguments

| | |
|---|---|
| `theta` | Starting values for the parameters of the loglikelhood function. |
| `loglkfn` | An inner function to compute the loglikelihood and its derivatives. The values returned by this function must be a list with `loglk`, `score` and `inf`. |
| `messg` | A labelling string to be printed as a part of warnings etc. Useful with nested calls to `mlefn`. |
| `niter` | Maximum number of iterations used. The default is 20. |
| `tol` | Level of tolerance for checking the convergence. |
| `guide` | Specification of the direction for convergence with the following choices:<br>`"uphill"` – seek a maximum;<br>`"downhill"` – seek a minimum;<br>`"no"` – straight Newton approach without using loglikelihood values to guide the search.<br>`"auto"` – only used in `mlefn.control.inner` when the inner function `loglkfn` requires a call to `mlefn` itself and we want it to determine and supply a legitmate value for that (inner) call. |
| `print.progress` | A numeric value used to control the printing of error messages (if any); 0 should be used if no printing is required. |
| `max.eigenrat` | An argument used in the inner function `greenstadt.step` to control the eigenvalues of the information matrix. This is the Greenstadt modification described in page 601 of Seber and Wild (1989). |
| `n.earlyit` | Number of iterations to be treated as `"early"`; The default is 0. |
| `constrain` | Specification of constrain on the parameter estimates with the following choices:<br>`"no"` – no constrains;<br>`"fix"` – fix some of the parameters at their starting values; |
| `fixed` | A vector specifying the parameters to be fixed, indicated by their orders in `theta`. Used only if `constrain="fix"`. |
| `Aconstrain` | an `I` matrix with number of rows equal to the number of "fixed" parameters. Used only if `constrain="fix"`. |
| `cconstrain` | A vector specifying the values of those "fixed" parameters.<br>Used only if `constrain="fix"`. |
| `control` | to pass control options |
| `...` | Further arguments passed to or from related functions. |

## Details

This is the base function to maximise, minimise or find stationary values for `theta` using the provided `loglkfn` function. All semi-parametric maximum likelihood approaches we have proposed in `missreg` library require this function to obtain maximum likelihood estimates of parameters. See "Description of the `missreg` Library" for more details.

### Value

A list with the following components.

| | |
|---|---|
| `theta` | Updated parameter estimates at this iteration. |
| `loglk` | Log-likelihood obtained from the current `theta` values. |
| `score` | Score vector obtained from the current `theta` values. |
| `inf` | Observed information matrix obtained from the curent `theta` values. |
| `constrscore` | Constrained score vector if `constrain="fix"`; otherwise `NULL`. |
| `constrinf` | Constrained observed information matrix if `constrain="fix"`; otherwise `NULL`. |
| `counter` | Number of iterations performed. |
| `error` | A numeric value indicating the types of errors during iterations; a value of 0 indicates no error. |

### Author(s)

Chris Wild, Yannan Jiang

### References

Nonlinear Regression, Seber and Wild, 1989. Wiley: New York.
Description of the `missreg` Library, Wild and Jiang, 2007.

---

## `MLInf` *Likelihood calculation function for the discrete partition version with stratification*

### Description

An outer function of `ML2Inf` to provide the value, score vector and information matrix at `theta` for the so-called profile loglikelihood `l_P(theta)` of the form `l*(theta, Q)` with strtified two-phase sampled data. It reduces to an unstratified approach when nstrata=1.

### Usage

```
MLInf(theta, nderivs = 2, ProspModInf, StratModInf, x, y,
      Aposn, Acounts, Bposn, Bcounts, rmat, Qmat,
      xStrat = rep(1, dim(x)[1]), extra = NULL,
      off.set = matrix(0, dim(x)[1], dim(x)[3]),
      control.inner = mlefn.control.inner(...), ...)
```

## Arguments

| | |
|---|---|
| `theta` | Starting values for parameter `theta` in the regression model. See `ML2Inf` for details. |
| `nderivs` | Number of derivatives to be calculated. |
| `ProspModInf` | See `ML2Inf`. |
| `StratModInf` | See `ML2Inf`. |
| `x` | See `ML2Inf`. |
| `y` | See `ML2Inf`. |
| `Aposn` | See `ML2Inf`. |
| `Acounts` | See `ML2Inf`. |
| `Bposn` | See `ML2Inf`. |
| `Bcounts` | See `ML2Inf`. |
| `rmat` | The `r_k^(s)` provided in a matrix form (K*S) with S the number of strata and K number of distinct `h`-values observed. |
| `Qmat` | The `Q_k^(s)` provided in a matrix form (K*S). |
| `xStrat` | A vector of values 1 to S specifying the stratum membership of each observation. |
| `extra` | Provides Qmat from last iteration as starting values for next inner iterative loop in `mlefn` function call. |
| `off.set` | See `ML2Inf`. |
| `control.inner` | See `ML2Inf`. |
| `...` | Further arguments passed to or from related functions. |

## Details

This is the direct function called by `mlefn` to calculate the value, score vector and observed information matrix at `theta` for the so-called profile loglikelihood `l_P(theta)` using the discrete partition version. It calls the inner function `ML2Inf` to evaluate `l*^(s)(theta,Q^(s))` within each s-stratum.

## Value

A list with the following components.

| | |
|---|---|
| `loglk` | Log-likelihood obtained from the current `theta` values. |
| `score` | Score vector obtained from the current `theta` values when `nderivs>=1`; `NULL` otherwise. |
| `inf` | Observed information matrix obtained from the current `theta` values when `nderivs=2`; `NULL` otherwise. |
| `extra` | A list providing updated Qmat values from previous iteration. |

## Author(s)

Chris Wild, Yannan Jiang

### References

Description of the `missreg` Library, Wild and Jiang, 2007.

### See Also

`ML2Inf`; `mlefn`

---

## `rclusbin` *Random intercept model for clustered binary data*

### Description

Fits random intercept models to clustered binary data with the two-phase missingness structure.

### Usage

```
rclusbin(formula, data, weights=NULL, ClusInd=NULL, IntraClus=NULL, xstrata=NULL, ystrata=NULL,
obstype.name=”obstype”, NMat=NULL, xs.includes=FALSE, MaxInClus=NULL, rmsingletons=FALSE, retrosamp=”
gamma=NULL, nzval0=20, fit=TRUE, devcheck=FALSE, linkname=”logit”, start=NULL, Qstart=NULL, sigma=NULL
control=mlefn.control(...), control.inner=mlefn.control.inner(...), ...)
```

### Arguments

| | |
|---|---|
| `formula` | A symbolic description of the model to be fitted. |
| `data` | A data frame containing all the variables required for analysis, including those for `xstrata`, `ystrata` and `obstype.name`. |
| `weights` | An optional vector of weights to be used in the fitting process. Should be `NULL` or the name of a numeric vector in the data frame. It provides weights at the individual level when there are clusters of size greater than one.<br>When all clusters are of size one, it provides weights at cluster=individual level. |
| `ClusInd` | Name of a vector in the data frame specifying cluster membership. Can be `NULL` if all clusters are of size one. |
| `IntraClus` | Name of a vector in the data frame specifying intra-cluster sequence of individual subjects in a cluster. The one with the smallest i.d. is treated as the proband who were originally sampled into a study. |
| `xstrata` | Specify names of the stratification variables to be used,e.g. `”vname”` or `c(”vname1”,”vname2”,...)`. Strata are defined by cross-classifiction of all levels.<br>This function only deals with the situation when clusters are defined within xstrata. |
| `ystrata` | Specify name of the variable defing the Y-strata. Compulsory when gamma probabilities are used (see Details for more descriptions). |
| `obstype.name` | Name of the variable specifying labels for observations by sampling and variable type: `”uncond”`, `”retro”`, `”xonly”`, `”y\|x”` or `”strata”`. |

| | |
|---|---|
| `NMat` | Population counts in a matrix form with rows and columns corresponding to Y-strata and X-strata respectively. Should not be provided when there is any observation of the type `"strata"`. |
| `xs.includes` | `TRUE` if `weights` specified for obervations labelled as "strata" include those observed at the second phase (i.e. `"retro"` or `"uncond"` observations). |
| `MaxInClus` | A value specifying the maximum number of individuals allowed in a cluster. Set to `NULL` if there is no limit. |
| `rmsingletons` | If `TRUE`, remove clusters of size one. |
| `retrosamp` | Three restrospective sampling schemes can be applied based on the Y-status of all subjects in the same cluster: `"proband"`, `"allcontrol"` and `"gamma"` (see Details for more descriptions). The default is `"proband"`. |
| `gamma` | A vector of length 2 specifying the probabilities that individuals belong to `Y=1` based on their cluster status (see Details for more descriptions). |
| `nzval0` | Number of points to calculate the zeros and weights needed for Gauss-Hermite quadrature. The default is 20. |
| `fit` | If `FALSE`, only stratum report will be generated without model fitting. |
| `devcheck` | If `TRUE`, check the first and second derivatives. The default should be `FALSE`. |
| `linkname` | A specification for the model link function. Three choices are provide: `"logit"`, `"probit"` or "cloglog". The default is `"logit"`. |
| `start` | Starting values for the regression parameters. The first p-coefficients are parameters for X-variables. The last parameter is for the random intercept term and normally denoted as `w=log(sigma)`.<br>The program cannot provide starting values for all data strctures so will force you to use this whenever it is necessary. |
| `Qstart` | An optional starting matrix for Pr(Ystratum=i\|Xstratum=j). Can be compulsory if the program cannot produce a valid starting value at some situations. |
| `sigma` | An optional starting value for `sigma`. The default value (when set to `NULL`) is 0.5. |
| `control` | Specify control parameters for the iterations in `mlefn` call. |
| `control.inner` | Specify control parameters for inner iterations nested within `mlefn` call. |
| `...` | Further arguments passed to or from related function. |

## Details

This function fits binary regression models with a random intercept of the form `a_i=e^{w*eps_i}` where `w=log(sigma)` and `eps_i` is standard normal for each cluster, along with a linear predictor `eta_{ij}=x_{ij}^T*beta` for the subject `j` in the `i^th` cluster.

The function can be applied to both prospective and retrospective data with various types of observations collected at different two-phase sampling schemes. Three retrospective samplings are considered with the Y-strata defined as:
(1) the case-control status of the proband only (`"proband"`); (2) the case-cotnrol status of all members in the same cluster (`"allcontrol"`). If any one of the members are cases, the cluster belongs to Y-strata=1 and otherwise Y-strata=0;

(3) the case-control status of all members in the same cluster plus the gamma probabilities (`"gamma"`). The conditional probability of Y-strata=1 depends on `sum_j{Y_j}=1` (with `gamma_1` probability) or `sum_j{Y_j}>1` (with `gamma_2` probability).
Here `Y_j` indicates case-control status (1 for a case and 0 for a control) of the `j^{th}` individual in a cluster.

### Source

http://www.stat.auckland.ac.nz/~wild

### References

Description of the `missreg` Library, Wild and Jiang, 2007.

### See Also

`ghq`, `rclusbin2`

### Examples

```
data(brainpairs)
brainpairs$obstype <- rep("retro", dim(brainpairs)[1])
z2 <- rclusbin(bt ~ ep + ca, ClusInd="id", IntraClus="relid", data=brainpairs)
summary(z2)

data(rdat00)
z3 <- rclusbin(y~x, ClusInd="cluster", data=rdat00, retrosamp="allcontrol")
summary(z3)
```

---

## `rclusbin2` *Random intercept model for clustered binary data following case-control sampling.*

### Description

Fits random intercept models to clustered binary data after case and control sampling, wherein interest is in the relationship between a binary response (Y) that is related to the sampling variable (Z).

### Usage

```
rclusbin2(formula1, formula2, weights=NULL, ClusInd.name=NULL, IntraClus.name=NULL, yname, xstrata=NU
obstype.name="obstype", data, NMat=NULL, xs.includes=FALSE, MaxInClus=NULL, rmsingletons=FALSE, retro
nzval0=20, fit=TRUE, devcheck=FALSE, linkname="logit", start=NULL, Qstart=NULL, sigma=NULL, paruse="xi
control=mlefn.control(...), control.inner=mlefn.control.inner(...), ...)
```

## Arguments

| | |
|---|---|
| `formula1` | A symbolic description of the random intercept model to be fitted, i.e. the model of interest. |
| `formula2` | A symbolic description of the auxiliary model to be fitted, between the sampling (case-control) variable and the binary response of interest. |
| `weights` | An optional vector of weights to be used in the fitting process. Should be `NULL` or the name of a numeric vector in the data frame. It provides weights at the individual level when there are clusters of size greater than one.<br>When all clusters are of size one, it provides weights at cluster=individual level. |
| `ClusInd.name` | Name of a vector in the data frame specifying cluster membership. Can be `NULL` if all clusters are of size one. |
| `IntraClus.name` | Name of a vector in the data frame specifying intra-cluster sequence of individual subjects in a cluster. The one with the smallest i.d. is treated as the proband who were originally sampled into a study. |
| `yname` | Name of the binary response variable of interest in the data frame. Must be specified. |
| `xstrata` | Specify names of the stratification variables to be used, e.g. `"vname"` or `c("vname1","vname2",...)`. Strata are defined by cross-classifiction of all levels.<br>This function only deals with the situation when clusters are defined within xstrata. |
| `ystrata` | Specify name of the variable defining the case and control strata. |
| `obstype.name` | Name of the variable specifying labels for observations by sampling and variable type: `"uncond"`, `"retro"`, `"xonly"`, `"y\|x"` or `"strata"`. |
| `data` | A data frame containing all the variables required for analysis, including those for `xstrata`, `ystrata` and `obstype.name`. |
| `NMat` | Population counts in a matrix form with rows and columns corresponding to case-control strata and X-strata respectively. Should not be provided when there is any observation of the type `"strata"`. |
| `xs.includes` | `TRUE` if `weights` specified for obervations labelled as "strata" include those observed at the second phase (i.e. `"retro"` or `"uncond"` observations). |
| `MaxInClus` | A value specifying the maximum number of individuals allowed in a cluster. Set to `NULL` if there is no limit. |
| `rmsingletons` | If `TRUE`, remove clusters of size one. |
| `retrosamp` | As the default, must be `TRUE` here. |
| `nzval0` | Number of points to calculate the zeros and weights needed for Gauss-Hermite quadrature. The default is 20. |
| `fit` | If `FALSE`, only stratum report will be generated without model fitting. |
| `devcheck` | If `TRUE`, check the first and second derivatives. The default should be `FALSE`. |
| `linkname` | A specification for the model link function. Three choices are provide: `"logit"`, `"probit"` or "cloglog". The default is `"logit"`. |
| `start` | Starting values for all regression parameters. |

| | |
|---|---|
| `Qstart` | An optional starting matrix for Pr(Zstratum=i\|Xstratum=j). Can be compulsory if the program cannot produce a valid starting value at some situations. |
| `sigma` | An optional starting value for `sigma`. The default value (when set to `NULL`) is 0.5. |
| `paruse` | As the default, must be `"xis"` here. |
| `control` | Specify control parameters for the iterations in `mlefn` call. |
| `control.inner` | Specify control parameters for inner iterations nested within `mlefn` call. |
| `...` | Further arguments passed to or from related function. |

## Details

To be added.

## Source

http://www.stat.auckland.ac.nz/~wild

## References

Longitudinal Studies of Binary Response Data Following Case-Control and Stratified Case-Control Sampling: Design and Analysis, Schildcrout and Rathouz, BIOMETRICS 2009.

## See Also

`ghq`, `rclusbin`

## Examples

```
data(adhd)
head(adhd)

adhd$obstype <- rep("retro", dim(adhd)[1])
adhd$probandS <- 2 - adhd$proband #as 1/2 for case/control
adhd$sexF <- adhd$sex-1 #as 1/0 for female/male
adhd$wave1 <- ifelse(adhd$wave==1, 1, 0)
adhd$wave2 <- ifelse(adhd$wave==2, 1, 0)

adhd1 <- adhd[adhd$wave==1,]
z0 <- glm(proband ~ adhd, family=binomial, data=adhd1)
z0$coefficients

nMat <- ftable(adhd1$sex~adhd1$probandS)  # 1=male; 2=female;
nMat

### Samping ratios for boys/girls (Schildcrout & Rathouz)
pi_ctF <- 1/22.6
pi_ctM <- 1/22.4
NMat <- cbind(c(113, 96/pi_ctM), c(25, 21/pi_ctF))

z <- rclusbin2(adhd ~ wave1+wave2+wave+sexF+african+other+wave*sexF+wave*african, proband.1~adhd.1, ClusInd.name
```

```
yname=”adhd”, ystrata=”probandS”, xstrata=”sex”, data=adhd, NMat=NMat, nzval0=40, control=mlefn.control(niter=100

summary(z)
```

# Index