

# Package ‘misaem’

July 5, 2020

**Title** Linear Regression and Logistic Regression with Missing  
Covariates

**Version** 1.0.0

**Date** 2020-07-04

**Description** Estimate parameters of linear regression and logistic regression with missing covariates with missing data, perform model selection and prediction, using EM-type algorithms.

**Depends** R (>= 3.4.0)

**Encoding** UTF-8

**License** GPL-3

**URL** <https://github.com/wjiang94/misaem.git>

**Imports** mvtnorm, stats, MASS, mice, norm, methods

**Suggests** knitr, rmarkdown

**LazyData** true

**VignetteBuilder** knitr

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Author** Wei Jiang [aut, cre],  
Pavlo Mozharovskyi [ctb]

**Maintainer** Wei Jiang <wei.jiang@polytechnique.edu>

**Repository** CRAN

**Date/Publication** 2020-07-05 11:50:03 UTC

## R topics documented:

combinations . . . . .	2
imputeEIIIP . . . . .	3
likelihood_saem . . . . .	3
log_reg . . . . .	4
louis_lr_saem . . . . .	5
miss.glm . . . . .	6

miss.glm.control . . . . .	8
miss.glm.fit . . . . .	9
miss.glm.model.select . . . . .	10
miss.lm . . . . .	11
miss.lm.control . . . . .	12
miss.lm.fit . . . . .	13
miss.lm.model.select . . . . .	14
predict.miss.glm . . . . .	15
predict.miss.lm . . . . .	16
print.miss.glm . . . . .	17
print.miss.lm . . . . .	18
print.summary.miss.glm . . . . .	18
print.summary.miss.lm . . . . .	19
summary.miss.glm . . . . .	19
summary.miss.lm . . . . .	20

<b>Index</b>	<b>21</b>
--------------	-----------

---

combinations	<i>combinations</i>
--------------	---------------------

---

## Description

Given all the possible patterns of missingness.

## Usage

```
combinations(p)
```

## Arguments

p                    Dimension of covariates.

## Value

A matrix containing all the possible missing patterns. Each row indicates a pattern of missingness. "1" means "observed", 0 means "missing".

## Examples

```
comb = combinations(5)
```

---

imputeE11P	<i>Function for imputing single point for linear regression model</i>
------------	---

---

**Description**

Function for imputing single point for linear regression model

**Usage**

```
imputeE11P(point, Sigma.inv)
```

**Arguments**

point	A single observation containing missing values.
Sigma.inv	Inverse of estimated $\Sigma$ .

**Value**

Imputed observation.

---

likelihood_saem	<i>likelihood_saem</i>
-----------------	------------------------

---

**Description**

Used in main function miss.saem. Calculate the observed log-likelihood for logistic regression model with missing data, using Monte Carlo version of Louis formula.

**Usage**

```
likelihood_saem(
  beta,
  mu,
  Sigma,
  Y,
  X.obs,
  rindic = as.matrix(is.na(X.obs)),
  whichcolXmissing = (1:ncol(rindic))[apply(rindic, 2, sum) > 0],
  mc.size = 2
)
```

**Arguments**

beta	Estimated parameter of logistic regression model.
mu	Estimated parameter $\mu$ .
Sigma	Estimated parameter $\Sigma$ .
Y	Response vector $N \times 1$
X.obs	Design matrix with missingness $N \times p$
rindic	Missing pattern of X.obs. If a component in X.obs is missing, the corresponding position in rindic is 1; else 0.
whichcolXmissing	The column index in covariate containing at least one missing observation.
mc.size	Monte Carlo sampling size.

**Value**

Observed log-likelihood.

**Examples**

```
# Generate dataset
N <- 50 # number of subjects
p <- 3 # number of explanatory variables
mu.star <- rep(0,p) # mean of the explanatory variables
Sigma.star <- diag(rep(1,p)) # covariance
beta.star <- c(1, 1, 0) # coefficients
beta0.star <- 0 # intercept
beta.true = c(beta0.star,beta.star)
X.complete <- matrix(rnorm(N*p), nrow=N)%%chol(Sigma.star) +
  matrix(rep(mu.star,N), nrow=N, byrow = TRUE)
p1 <- 1/(1+exp(-X.complete%*%beta.star-beta0.star))
y <- as.numeric(runif(N)<p1)
# Generate missingness
p.miss <- 0.10
patterns <- runif(N*p)<p.miss #missing completely at random
X.obs <- X.complete
X.obs[patterns] <- NA

# Observed log-likelihood
ll_obs = likelihood_saem(beta.true,mu.star,Sigma.star,y,X.obs)
```

---

log\_reg

log\_reg

---

**Description**

Calculate the likelihood or log-likelihood for one observation of logistic regression model .

**Usage**

```
log_reg(y, x, beta, iflog = TRUE)
```

**Arguments**

y	Response value (0 or 1).
x	Covariate vector of dimension $p \times 1$ .
beta	Estimated parameter of logistic regression model.
iflog	If TRUE, log_reg calculate the log-likelihood; else likelihood.

**Value**

Likelihood or log-likelihood.

**Examples**

```
res = log_reg(1,c(1,2,3),c(1,-1,1))
```

---

louis_lr_saem	<i>louis_lr_saem</i>
---------------	----------------------

---

**Description**

Used in main function miss.saem. Calculate the variance of estimated parameters for logistic regression model with missing data, using Monte Carlo version of Louis formula.

**Usage**

```
louis_lr_saem(
  beta,
  mu,
  Sigma,
  Y,
  X.obs,
  pos_var = 1:ncol(X.obs),
  rindic = as.matrix(is.na(X.obs)),
  whichcolXmissing = (1:ncol(rindic))[apply(rindic, 2, sum) > 0],
  mc.size = 2
)
```

**Arguments**

beta	Estimated parameter of logistic regression model.
mu	Estimated parameter $\mu$ .
Sigma	Estimated parameter $\Sigma$ .
Y	Response vector $N \times 1$
X.obs	Design matrix with missingness $N \times p$
pos_var	Index of selected covariates.
rindic	Missing pattern of X.obs. If a component in X.obs is missing, the corresponding position in rindic is 1; else 0.
whichcolXmissing	The column index in covariate containing at least one missing observation.
mc.size	Monte Carlo sampling size.

**Value**

Variance of estimated  $\beta$ .

**Examples**

```
# Generate dataset
N <- 50 # number of subjects
p <- 3 # number of explanatory variables
mu.star <- rep(0,p) # mean of the explanatory variables
Sigma.star <- diag(rep(1,p)) # covariance
beta.star <- c(1, 1, 0) # coefficients
beta0.star <- 0 # intercept
beta.true = c(beta0.star,beta.star)
X.complete <- matrix(rnorm(N*p), nrow=N)%*%chol(Sigma.star) +
  matrix(rep(mu.star,N), nrow=N, byrow = TRUE)
p1 <- 1/(1+exp(-X.complete%*%beta.star-beta0.star))
y <- as.numeric(runif(N)<p1)
# Generate missingness
p.miss <- 0.10
patterns <- runif(N*p)<p.miss #missing completely at random
X.obs <- X.complete
X.obs[patterns] <- NA

# Louis formula to obtain variance of estimates
V_obs = louis_lr_saem(beta.true,mu.star,Sigma.star,y,X.obs)
```

**Description**

This function is used to perform statistical inference for logistic regression model with missing values, by algorithm SAEM.

**Usage**

```
miss.glm(formula, data, control = list(...), ...)
```

**Arguments**

formula	an object of class "formula": a symbolic description of the logistic regression model to be fitted.
data	an optional data frame containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which miss.glm is called.
control	a list of parameters for controlling the fitting process. For miss.glm.fit this is passed to <code>miss.glm.control</code> .
...	arguments to be used to form the default control argument if it is not supplied directly.

**Value**

An object of class "miss.glm": a list with following components:

coefficients	Estimated $\beta$ .
ll	Observed log-likelihood.
var.covar	Variance-covariance matrix for estimated parameters.
s.err	Standard error for estimated parameters.
mu.X	Estimated $\mu$ .
Sig.X	Estimated $\Sigma$ .
call	the matched call.
formula	the formula supplied.

**Examples**

```
# Generate dataset
N <- 100 # number of subjects
p <- 3   # number of explanatory variables
mu.star <- rep(0,p) # mean of the explanatory variables
Sigma.star <- diag(rep(1,p)) # covariance
beta.star <- c(1, 1, 0) # coefficients
beta0.star <- 0 # intercept
beta.true = c(beta0.star,beta.star)
X.complete <- matrix(rnorm(N*p), nrow=N)%%chol(Sigma.star) +
  matrix(rep(mu.star,N), nrow=N, byrow = TRUE)
p1 <- 1/(1+exp(-X.complete%*%beta.star-beta0.star))
y <- as.numeric(runif(N)<p1)
```

```

# Generate missingness
p.miss <- 0.10
patterns <- runif(N*p)<p.miss #missing completely at random
X.obs <- X.complete
X.obs[patterns] <- NA

df.obs = data.frame(y,X.obs)

# SAEM
miss.list = miss.glm(y~., data=df.obs, print_iter=FALSE,seed=100)
print(miss.list)
print(summary(miss.list))
summary(miss.list)$coef

```

---

miss.glm.control      *Auxiliary for Controlling Fitting*

---

## Description

Auxiliary function for `miss.glm` fitting. Typically only used internally by `miss.glm.fit`.

## Usage

```

miss.glm.control(
  maxruns = 500,
  tol_em = 1e-07,
  nmcmc = 2,
  tau = 1,
  k1 = 50,
  subsets = NA,
  seed = NA,
  print_iter = TRUE,
  var_cal = TRUE,
  ll_obs_cal = TRUE
)

```

## Arguments

<code>maxruns</code>	maximum number of iterations. The default is <code>maxruns = 500</code> .
<code>tol_em</code>	the tolerance to stop SAEM. The default is <code>tol_em = 1e-7</code> .
<code>nmcmc</code>	the MCMC length. The default is <code>nmcmc = 2</code> .
<code>tau</code>	rate $\tau$ in the step size $(k - k_1)^{-\tau}$ . The default is <code>tau = 1</code> .
<code>k1</code>	number of first iterations $k_1$ in the step size $(k - k_1)^{-\tau}$ . The default is <code>k1=50</code> .
<code>subsets</code>	Index of selected covariates if any. The default is all the covariates.
<code>seed</code>	an integer as a seed set for the random generator.



print_iter	logical indicating if output should be produced for each iteration.
var_cal	logical indicating if the variance of estimated parameters should be calculated.
ll_obs_cal	logical indicating if the observed log-likelihood should be calculated.

**Value**

A list with components named as the arguments.

**Examples**

```
## For examples see example(miss.glm)
```

---

miss.glm.fit	<i>Fitting Logistic Regression Models with Missing Values</i>
--------------	---

---

**Description**

This function is used inside `miss.glm` to fit logistic regression model with missing values, by algorithm SAEM.

**Usage**

```
miss.glm.fit(x, y, control = list())
```

**Arguments**

x	design matrix with missingness $N \times p$ .
y	response vector $N \times 1$ .
control	a list of parameters for controlling the fitting process. For <code>miss.glm.fit</code> this is passed to <code>miss.glm.control</code> .

**Value**

a list with following components:

coefficients	Estimated $\beta$ .
ll	Observed log-likelihood.
var.covar	Variance-covariance matrix for estimated parameters.
s.err	Standard error for estimated parameters.
mu.X	Estimated $\mu$ .
Sig.X	Estimated $\Sigma$ .

**Examples**

```
## For examples see example(miss.glm)
```

---

miss.glm.model.select *miss.glm.model.select*

---

## Description

Model selection for the logistic regression model with missing data.

## Usage

```
miss.glm.model.select(Y, X, seed = NA)
```

## Arguments

Y	Binary response vector $N \times 1$
X	Design matrix with missingness $N \times p$
seed	An integer as a seed set for the random generator. The default value is 200.

## Value

An object of class "miss.glm".

## Examples

```
# Generate dataset
N <- 40 # number of subjects
p <- 3 # number of explanatory variables
mu.star <- rep(0,p) # mean of the explanatory variables
Sigma.star <- diag(rep(1,p)) # covariance
beta.star <- c(1, 1, 0) # coefficients
beta0.star <- 0 # intercept
beta.true = c(beta0.star,beta.star)
X.complete <- matrix(rnorm(N*p), nrow=N)%%chol(Sigma.star) +
  matrix(rep(mu.star,N), nrow=N, byrow = TRUE)
p1 <- 1/(1+exp(-X.complete%*%beta.star-beta0.star))
Y <- as.numeric(runif(N)<p1)
# Generate missingness
p.miss <- 0.10
patterns <- runif(N*p)<p.miss #missing completely at random
X <- X.complete
X[patterns] <- NA
# model selection for SAEM
miss.model = miss.glm.model.select(Y,X,seed=100)
print(miss.model)
```

---

miss.lm                      *Statistical Inference for Linear Regression Models with Missing Values*

---

### Description

This function is used to perform statistical inference for linear regression model with missing values, by algorithm EM.

### Usage

```
miss.lm(formula, data, control = list(...), ...)
```

### Arguments

formula	an object of class " <code>formula</code> ": a symbolic description of the linear regression model to be fitted.
data	an optional data frame containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>miss.lm</code> is called.
control	a list of parameters for controlling the fitting process. For <code>miss.lm.fit</code> this is passed to <code>miss.lm.control</code> .
...	arguments to be used to form the default control argument if it is not supplied directly.

### Value

An object of class "`miss.lm`": a list with following components:

coefficients	Estimated $\beta$ .
ll	Observed log-likelihood.
s.resid	Estimated standard error for residuals.
s.err	Standard error for estimated parameters.
mu.X	Estimated $\mu$ .
Sig.X	Estimated $\Sigma$ .
call	the matched call.
formula	the formula supplied.

### Examples

```
# Generate complete data
set.seed(1)
mu.X <- c(1, 1)
Sigma.X <- matrix(c(1, 1, 1, 4), nrow = 2)
n <- 50
p <- 2
```

```

X.complete <- matrix(rnorm(n*p), nrow=n)%*%chol(Sigma.X) +
  matrix(rep(mu.X,n), nrow=n, byrow = TRUE)
b <- c(2, 3, -1)
sigma.eps <- 0.25
y <- cbind(rep(1, n), X.complete) %*% b + rnorm(n, 0, sigma.eps)

# Add missing values
p.miss <- 0.10
patterns <- runif(n*p)<p.miss #missing completely at random
X.obs <- X.complete
X.obs[patterns] <- NA

# Estimate regression using EM
df.obs = data.frame(y,X.obs)
miss.list = miss.lm(y~., data=df.obs)
print(miss.list)
print(summary(miss.list))
summary(miss.list)$coef

```

---

miss.lm.control

*Auxiliary for Controlling Fitting*


---

## Description

Auxiliary function for `miss.lm` fitting. Typically only used internally by `miss.lm.fit`.

## Usage

```
miss.lm.control(maxruns = 500, tol_em = 1e-07, print_iter = TRUE)
```

## Arguments

<code>maxruns</code>	maximum number of iterations. The default is <code>maxruns = 500</code> .
<code>tol_em</code>	the tolerance to stop EM. The default is <code>tol_em = 1e-4</code> .
<code>print_iter</code>	logical indicating if output should be produced for each iteration.

## Value

A list with components named as the arguments.

## Examples

```
## For examples see example(miss.lm)
```

---

`miss.lm.fit`*Fitting Linear Regression Model with Missing Values*

---

**Description**

This function is used inside `miss.lm` to fit linear regression model with missing values, by EM algorithm.

**Usage**

```
miss.lm.fit(x, y, control = list())
```

**Arguments**

<code>x</code>	design matrix with missingness $N \times p$ .
<code>y</code>	response vector $N \times 1$ .
<code>control</code>	a list of parameters for controlling the fitting process. For <code>miss.lm.fit</code> this is passed to <a href="#">miss.lm.control</a> .

**Value**

a list with following components:

<code>coefficients</code>	Estimated $\beta$ .
<code>ll</code>	Observed log-likelihood.
<code>s.resid</code>	Estimated standard error for residuals.
<code>s.err</code>	Standard error for estimated parameters.
<code>mu.X</code>	Estimated $\mu$ .
<code>Sig.X</code>	Estimated $\Sigma$ .

**Examples**

```
## For examples see example(miss.lm)
```

---

`miss.lm.model.select` *miss.lm.model.select*

---

**Description**

Model selection for the linear regression model with missing data.

**Usage**

```
miss.lm.model.select(Y, X)
```

**Arguments**

Y	Response vector $N \times 1$
X	Design matrix with missingness $N \times p$

**Value**

An object of class "miss.lm".

**Examples**

```
# Generate complete data
set.seed(1)
mu.X <- c(1, 1)
Sigma.X <- matrix(c(1, 1, 1, 4), nrow = 2)
n <- 50
p <- 2
X.complete <- matrix(rnorm(n*p), nrow=n)%%chol(Sigma.X) +
  matrix(rep(mu.X,n), nrow=n, byrow = TRUE)
b <- c(2, 0, -1)
sigma.eps <- 0.25
y <- cbind(rep(1, n), X.complete) %*% b + rnorm(n, 0, sigma.eps)

# Add missing values
p.miss <- 0.10
patterns <- runif(n*p)<p.miss #missing completely at random
X.obs <- X.complete
X.obs[patterns] <- NA

# model selection
miss.model = miss.lm.model.select(y, X.obs)
print(miss.model)
```

---

predict.miss.glm	<i>Prediction on test with missing values for the logistic regression model.</i>
------------------	--

---

### Description

Prediction on test with missing values for the logistic regression model.

### Usage

```
## S3 method for class 'miss.glm'
predict(object, newdata = NULL, seed = NA, method = "map", ...)
```

### Arguments

object	a fitted object of class inheriting from "miss.glm".
newdata	a data frame in which to look for variables with which to predict. It can contain missing values.
seed	An integer as a seed set for the random generator.
method	The name of method to deal with missing values in test set. It can be 'map'(maximum a posteriori) or 'impute' (imputation by conditional expectation). Default is 'map'.
...	Further arguments passed to or from other methods.

### Value

pr.saem	The prediction result for logistic regression: the probability of response $y=1$ .
---------	--

### Examples

```
# Generate dataset
N <- 100 # number of subjects
p <- 3   # number of explanatory variables
mu.star <- rep(0,p) # mean of the explanatory variables
Sigma.star <- diag(rep(1,p)) # covariance
beta.star <- c(1, 1, 0) # coefficients
beta0.star <- 0 # intercept
beta.true = c(beta0.star,beta.star)
X.complete <- matrix(rnorm(N*p), nrow=N)%%chol(Sigma.star) +
  matrix(rep(mu.star,N), nrow=N, byrow = TRUE)
p1 <- 1/(1+exp(-X.complete%*%beta.star-beta0.star))
y <- as.numeric(runif(N)<p1)

# Generate missingness
p.miss <- 0.10
patterns <- runif(N*p)<p.miss #missing completely at random
X.obs <- X.complete
X.obs[patterns] <- NA
```

```

df.obs = data.frame(y,X.obs)

# SAEM
miss.list = miss.glm(y~., data=df.obs, print_iter=FALSE,seed=100)

# Generate new dataset for prediction
Nt <- 20
Xt <- matrix(rnorm(Nt*p), nrow=Nt)%*%chol(Sigma.star)+
  matrix(rep(mu.star,Nt), nrow=Nt, byrow = TRUE)
# Generate missingness in new dataset
patterns <- runif(Nt*p)<p.miss
Xt.obs <- Xt
Xt.obs[patterns] <- NA

# Prediction with missing values
miss.prob = predict(miss.list, data.frame(Xt.obs), method='map')
print(miss.prob)

```

---

predict.miss.lm	<i>Prediction on test with missing values for the logistic regression model.</i>
-----------------	--

---

## Description

Prediction on test with missing values for the logistic regression model.

## Usage

```

## S3 method for class 'miss.lm'
predict(object, newdata = NULL, seed = NA, ...)

```

## Arguments

object	a fitted object of class inheriting from "miss.lm".
newdata	a data frame in which to look for variables with which to predict. It can contain missing values.
seed	An integer as a seed set for the random generator.
...	Further arguments passed to or from other methods.

## Value

pr.y	The prediction result for linear regression.
------	--



**Examples**

```

# Generate complete data
set.seed(1)
mu.X <- c(1, 1)
Sigma.X <- matrix(c(1, 1, 1, 4), nrow = 2)
n <- 50 # train set size
p <- 2 # number of covariates
X.complete <- matrix(rnorm(n*p), nrow=n)%*%chol(Sigma.X) +
  matrix(rep(mu.X,n), nrow=n, byrow = TRUE)

b <- c(2, 3, -1)
sigma.eps <- 0.25
y <- cbind(rep(1, n), X.complete) %*% b +
  rnorm(n, 0, sigma.eps)

# Add missing values
p.miss <- 0.10
patterns <- runif(n*p)<p.miss #missing completely at random
X.obs <- X.complete
X.obs[patterns] <- NA
# Estimate regression using EM
df.obs = data.frame(y ,X.obs)
miss.list = miss.lm(y~., data=df.obs)

# Generate new dataset for prediction
nt <- 20
Xt <- matrix(rnorm(nt*p), nrow=nt)%*%chol(Sigma.X)+
  matrix(rep(mu.X,nt), nrow=nt, byrow = TRUE)
# Generate missingness in new dataset
patterns <- runif(nt*p)<p.miss
Xt.obs <- Xt
Xt.obs[patterns] <- NA

# Prediction with missing values
miss.pred = predict(miss.list, data.frame(Xt.obs))
print(miss.pred)

```

---

print.miss.glm

*Print miss.glm*


---

**Description**

Print results for class `miss.glm`.

**Usage**

```

## S3 method for class 'miss.glm'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

```

**Arguments**

x                    an object of class "miss.glm", usually, a result of a call to [miss.glm](#).  
 digits                minimal number of significant digits.  
 ...                    further arguments passed to or from other methods.

**Examples**

```
## For examples see example(miss.glm)
```

---

```
print.miss.lm            Print miss.lm
```

---

**Description**

Print results for class `miss.lm`.

**Usage**

```
## S3 method for class 'miss.lm'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

**Arguments**

x                    an object of class "miss.lm", usually, a result of a call to [miss.lm](#).  
 digits                minimal number of significant digits.  
 ...                    further arguments passed to or from other methods.

**Examples**

```
## For examples see example(miss.lm)
```

---

```
print.summary.miss.glm            Print Summary of miss.glm
```

---

**Description**

Print results for class `summary.miss.glm`.

**Usage**

```
## S3 method for class 'summary.miss.glm'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

### Arguments

x                    an object of class "summary.miss.glm", usually, a result of a call to [summary.miss.glm](#).  
digits                minimal number of significant digits.  
...                   further arguments passed to or from other methods.

### Examples

```
## For examples see example(miss.glm)
```

---

`print.summary.miss.lm` *Print Summary of miss.lm*

---

### Description

Print results for class `summary.miss.lm`.

### Usage

```
## S3 method for class 'summary.miss.lm'  
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

### Arguments

x                    an object of class "summary.miss.lm", usually, a result of a call to [summary.miss.lm](#).  
digits                minimal number of significant digits.  
...                   further arguments passed to or from other methods.

### Examples

```
## For examples see example(miss.lm)
```

---

`summary.miss.glm`        *Summarizing Fits for miss.glm*

---

### Description

Summary for class `miss.glm`.

### Usage

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

**Arguments**

object            an object of class "miss.glm", usually, a result of a call to [miss.glm](#).  
 ...              Further arguments passed to or from other methods.

**Value**

An object of class "summary.miss.glm", a list with following components:

coefficients    The matrix of coefficients and standard errors  
 loglikelihood   Observed log-likelihood.  
 call            the component from object.  
 formula        the component from object.

**Examples**

```
## For examples see example(miss.glm)
```

---

summary.miss.lm	<i>Summarizing Fits for miss.lm</i>
-----------------	-------------------------------------

---

**Description**

Summary for class miss.lm.

**Usage**

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

**Arguments**

object            an object of class "miss.lm", usually, a result of a call to [miss.lm](#).  
 ...              Further arguments passed to or from other methods.

**Value**

An object of class "summary.miss.lm", a list with following components:

coefficients    The matrix of coefficients and standard errors.  
 loglikelihood   Observed log-likelihood.  
 call            the component from object.  
 formula        the component from object.

**Examples**

```
## For examples see example(miss.lm)
```

# Index

combinations, [2](#)

formula, [7](#), [11](#)

imputeEllP, [3](#)

likelihood\_saem, [3](#)

log\_reg, [4](#)

louis\_lr\_saem, [5](#)

miss.glm, [6](#), [8](#), [18](#), [20](#)

miss.glm.control, [7](#), [8](#), [9](#)

miss.glm.fit, [8](#), [9](#)

miss.glm.model.select, [10](#)

miss.lm, [11](#), [12](#), [18](#), [20](#)

miss.lm.control, [11](#), [12](#), [13](#)

miss.lm.fit, [12](#), [13](#)

miss.lm.model.select, [14](#)

predict.miss.glm, [15](#)

predict.miss.lm, [16](#)

print.miss.glm, [17](#)

print.miss.lm, [18](#)

print.summary.miss.glm, [18](#)

print.summary.miss.lm, [19](#)

summary.miss.glm, [19](#), [19](#)

summary.miss.lm, [19](#), [20](#)