

Package ‘ILSE’

January 20, 2025

Type Package

Title Linear Regression Based on 'ILSE' for Missing Data

Version 1.1.7

License GPL-3

Author Wei Liu [aut, cre],
Huazhen Lin [aut],
Wei Lan [aut]

Maintainer Wei Liu <weiliu@smail.swufe.edu.cn>

Description Linear regression when covariates include missing values by embedding the correlation information between covariates. Especially for block missing data, it works well. 'ILSE' conducts imputation and regression simultaneously and iteratively. More details can be referred to
Huazhen Lin, Wei Liu and Wei Lan. (2021) <[doi:10.1080/07350015.2019.1635486](https://doi.org/10.1080/07350015.2019.1635486)>.

URL <https://github.com/feiyong/ILSE>

BugReports <https://github.com/feiyong/ILSE/issues>

Encoding UTF-8

LazyData true

Date 2022-01-31

Depends R (>= 3.0.1)

Imports stats, Rcpp, pbapply

Suggests knitr, rmarkdown

LinkingTo Rcpp, RcppArmadillo

VignetteBuilder knitr

NeedsCompilation yes

RoxygenNote 7.1.1

Repository CRAN

Date/Publication 2022-01-31 03:10:05 UTC

Contents

Coef	2
cor.mat	3
cov.mat	4
fimlreg	5
ilse	6
kern	8
nhanes	9
print	10
summary	11
Index	13

Coef	<i>Extracts Regression Coefficients</i>
------	---

Description

extracts model coefficients from object of class "ilse".

Usage

```
Coef(object)
```

Arguments

object an object of class "ilse".

Value

Coefficients extracted from object.

See Also

coef, coefficient

Examples

```
# example one
data(nhanes)
NA1m2 <- ilse(age~., data=nhanes)
Coef(NA1m2)
```

`cor.mat`*Generate Two Type of Correlation Matrix*

Description

Generate two type of correlation matrix

Usage

```
cor.mat(p, rho, type='toeplitz')
```

Arguments

<code>p</code>	a positive integer, the dimension of correlation matrix.
<code>rho</code>	a value between 0 and 1, a baseline vlaue of correlation coefficient.
<code>type</code>	a character, specify the type of correlation matrix and only include 'toeplitz' and 'identity' in current version.

Details

The argument rho specify the size of correlation coefficient. As for argument type, if type='toeplitz', $\sigma_{ij}=\rho^{|i-j|}$; if type ='identity', $\sigma_{ij}=\rho$ when $i!=j$ and $\sigma_{ij}=1$ when $i=j$.

Value

return a correlation matrix with a type of specified structure.

Note

nothing

Author(s)

Liu Wei

References

nothing.

See Also

cov2cor

Examples

```
cor.mat(5, 0.5)
cor.mat(5, 0.5, type='identity')
```

`cov.mat`*Generate Two Type of Covariance Matrix*

Description

Generate two type of covariance matrix

Usage

```
cov.mat(sdvec, rho, type='toeplitz')
```

Arguments

<code>sdvec</code>	a positive vector, standard deviation of each random variable.
<code>rho</code>	a value between 0 and 1, a baseline vlaue of correlation coefficient.
<code>type</code>	a character, specify the type of correlation matrix and only include 'toeplitz' and 'identity' in current version.

Details

The argument rho specify the size of correlation coefficient. As for argument type, if type='toeplitz', $\sigma_{ij} = \rho^{|i-j|}$; if type = 'identity', $\sigma_{ij} = \rho$ when $i \neq j$ and $\sigma_{ij} = 1$ when $i = j$.

Value

return a covariance matrix with a type of specified structure.

Note

nothing

Author(s)

Liu Wei

References

nothing.

See Also

`cov2cor`

Examples

```
cov.mat(rep(5,5), 0.5)
cov.mat(c(2,4,3), 0.5, type='identity')
```

`fmlreg`*Full Information Maximum Likelihood Linear Regression*

Description

Estimate regression coefficients based on Full Information Maximum Likelihood Estimation, which can couple missing data, including response missing or covariates missing.

Usage

```
fmlreg(...)  
  
## S3 method for class 'formula'  
fmlreg(formula, data=NULL, ...)  
## S3 method for class 'numeric'  
fmlreg(Y, X, ...)
```

Arguments

<code>formula</code>	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
<code>Y</code>	a numeric vector, the response variable.
<code>X</code>	a numeric matrix that may include NAs, the covariate matrix.
<code>data</code>	an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a 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>else</code> is called.
<code>...</code>	Optional arguments.

Details

Note that arguments `...` of `stats::nlm` are the parameters of algorithm, see the details in help file of "nlm". "fmlreg" can cope with any type of missing data.

Value

Return a list including following components:

<code>beta</code>	A named vector of coefficients
<code>formula</code>	The formula used
<code>data</code>	The raw data

Author(s)

Liu Wei

See Also[ilse](#)**Examples**

```

data(nhanes)
## example one: include missing value
fiml1 <- fimlreg(age~., data=nhanes)
print(fiml1)
# example two: No missing vlaue
## example two: No missing value
n <- 100
group <- rnorm(n, sd=4)
weight <- 3.2*group + 1.5 + rnorm(n, sd=0.1)
fimllm <- fimlreg(weight~group, data=data.frame(weight=weight, group=group))
print(fimllm)

```

*ilse**Linear Regression by Iterative Least Square Estimation*

Description

Linear regression when covariates include missing values embedding the correlation information between covariates by Iterative Least Square Estimation.

Usage

```

ilse(...)
## S3 method for class 'formula'
ilse(formula, data=NULL, bw=NULL, k.type=NULL, method="Par.cond", ...)
## S3 method for class 'numeric'
ilse(Y, X,bw=NULL, k.type=NULL, method="Par.cond", max.iter=20,
     peps=1e-5, feps = 1e-7, arma=TRUE, verbose=FALSE, ...)

```

Arguments

...	Arguments passed to other methods.
formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
Y	a numeric vector, the reponse variable.
X	a numeric matrix that may include NAs, the covariate matrix.
data	an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a 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>ilse</code> is called.

<code>bw</code>	a positive value, specify the bandwidth in estimating missing values, default as NULL. When <code>bw=NULL</code> , it is automatically selected by empirical method.
<code>k.type</code>	an optional character string, specify the type of kernel used in iterative estimating algorithm and support <code>'epk'</code> , <code>'biweight'</code> , <code>'triangle'</code> , <code>'gaussian'</code> , <code>'triweight'</code> , <code>'tricube'</code> , <code>'cosine'</code> , <code>'uniform'</code> in current version, default as <code>'gaussian'</code> .
<code>method</code>	an optional character string, specify the iterative algorithm, support <code>'Par.cond'</code> and <code>'Full.cond'</code> in current version.
<code>max.iter</code>	an optional positive integer, the maximum iterative times, default as <code>'20'</code> .
<code>peps</code>	an optional positive value, tolerance vlaue of relative variation rate of estimated parametric vector, default as <code>'1e-7'</code> .
<code>feps</code>	an optional positive vlaue, tolerance vlaue of relative variation rate of objective function value, default as <code>'1e-7'</code> .
<code>arma</code>	an optional logical value, whether use armadillo and Rcpp to speed computation, default as <code>TRUE</code>
<code>verbose</code>	an optional logical value, indicate whether output the iterative information, default as <code>'TRUE'</code> .

Details

Models for `ilse` are specified symbolically. A typical model has the form `response ~ terms` where `response` is the (numeric) response vector and `terms` is a series of terms which specifies a linear predictor for response. A terms specification of the form `first + second` indicates all the terms in `first` together with all the terms in `second` with duplicates removed. A specification of the form `first:second` indicates the set of terms obtained by taking the interactions of all terms in `first` with all terms in `second`. The specification `first*second` indicates the cross of `first` and `second`. This is the same as `first + second + first:second`.

Value

`ilse` returns an object of class `"ilse"`.

The functions `summary` and `anova` are used to obtain and print a summary and analysis of variance table of the results. The generic accessor functions `coefficients`, `effects`, `fitted.values` and `residuals` extract various useful features of the value returned by `lm`.

An object of class `"ilse"` is a list containing at least the following components:

<code>beta</code>	a named vector of coefficients
<code>hX</code>	a imputed design matrix
<code>d.fn</code>	a nonnegative value, vlaue of relative variation rate of objective function value
<code>d.par</code>	a nonnegative value, relative variation rate of estimated parametric vector when algorithm stopped.
<code>iterations</code>	a positive integer, iterative times in total.
<code>residuals</code>	the residuals, that is response minus fitted values.
<code>fitted.values</code>	the fitted mean values.
<code>inargs</code>	a list including all input arguments.

Note

nothing

Author(s)

Wei Liu

References

Huazhen Lin, Wei Liu, & Wei Lan (2021). Regression Analysis with individual-specific patterns of missing covariates. *Journal of Business & Economic Statistics*, 39(1), 179-188.

See Also

[lm](#)

Examples

```
## exmaple one: include missing value
data(nhanes)
NALm1 <- ilse(age~., data=nhanes,bw=1,
  method = 'Par.cond', k.type='gaussian', verbose = TRUE)
print(NALm1)
NALm2 <- ilse(age~., data=nhanes, method = 'Full.cond')
print(NALm2)
## example two: No missing value
n <- 100
group <- rnorm(n, sd=4)
weight <- 3.2*group + 1.5 + rnorm(n, sd=0.1)
NALm3 <- ilse(weight~group, data=data.frame(weight=weight, group=group),
  intercept = FALSE)
print(NALm3)
```

kern

Kernel Function

Description

Different type of kernel functions.

Usage

```
kern(u, type='epk')
```


Arguments

`u` a numeric vector, evaluated points in kernel function.

`type` a optional character string, specify the type of used kernel function and support 'epk', 'biweight', 'triangle', 'gaussian', 'triweight', 'tricube', 'cosine', 'uniform' in current version, default as 'epk'.

Details

Note that $K(u_i) = K(X_{i-x_0})$ where $u = (X_{1-x_0}, \dots, X_{n-x_0})$ and $K_h(u_i) = 1/h * K((X_{i-x_0})/h)$ where h is bandwidth.

Value

Return a numeric vector with length equal to 'u'.

Author(s)

Liu Wei

See Also

KernSmooth package

Examples

```
library(graphics)
u <- seq(-1,1,by=0.01)
(Ku <- kern(u))
plot(u, Ku, type='l')
# gaussian kernel
plot(u, kern(u, type='gaussian'), type='l')
# cosine kernel
plot(u, Ku <- kern(u, type='cosine'), type='l')
```

nhanes

NHANES example - all variables numerical

Description

A small data set with missing values.

Format

A data frame with 25 observations on the following 4 variables. age: Age group (1=20-39, 2=40-59, 3=60+).

bmi: Body mass index (kg/m**2).

hyp: Hypertensive (1=no,2=yes).

chl: Total serum cholesterol (mg/dL).

Details

A small data set with all numerical variables. The data set `nhanes2` is the same data set, but with `age` and `hyp` treated as factors.

Source

Schafer, J.L. (1997). *Analysis of Incomplete Multivariate Data*. London: Chapman & Hall. Table 6.14.

Examples

```
# example one
data(nhanes)
bw <- 1
ilse(age~., data=nhanes,bw=bw)
```

```
print
```

Print the Information of FIML or ILSE methods

Description

print method for class "ilse" or class "fiml".

Usage

```
print(object)
## S3 method for class 'ilse'
print(object)

## S3 method for class 'fiml'
print(object)
```

Arguments

`object` an object of class "ilse" or "fiml".

Value

For "ilse", print the basic information of ilse estimation and algorithm and return a list including

<code>beta</code>	a named vector of coefficients
<code>Bmat</code>	a named matrix that summary the estimated beta in every iteration.
<code>residuals</code>	the residuals, that is response minus fitted values.
<code>fitted.values</code>	the fitted mean values.
<code>d.fn</code>	a nonnegative value, vlaue of relative variation rate of objective function value

d.par a nonnegative value, relative variation rate of estimated parametric vector when algorithm stopped.

K a positive integer, iterative times in total.

For "fiml", print the basic information of fiml estimation and return a list including

beta A named vector of coefficients

iterations A positive integer, iterative times in total.

stop.code The stop code returned by nlm.

See Also

print.lm

Examples

```
data(nhanes)
NA1m1 <- ilse(age~., data=nhanes)
a <- print(NA1m1)
a

fimllm <- fimlreg(age~., data=nhanes, iterlim= 40)
b <- print(fimllm)
b
```

summary

Summarizing the inference information for ILSE or FIML methods

Description

summary method for class "ilse" or "fiml".

Usage

```
summary(object, Nbt=20)

## S3 method for class 'ilse'
summary(object, Nbt=20)

## S3 method for class 'fiml'
summary(object, Nbt=20)

##
Fitted.values(object)
##
Residuals(object)
```

Arguments

`object` an object of class "ilse".
`Nbt` an positive integer, the repeated times of bootstrap to eatimate covariance matrix of regression coefficient.

Value

The function `summary.ilse` computes and returns a named matrix of summary statistics of the fitted linear model given in `object` by ILSE or FIML methods. The function `Fitted.values` return a vector, fitted repsonse vlaues. The function `Residuals` return a vector, residuals.

See Also

`summary.lm` `fitted.vlaues` `residuals`

Examples

```
# example one
data(nhanes)
NALm <- ilse(age~., data=nhanes)
summary(NALm, Nbt=5)

fimllm <- fimlreg(age~., data=nhanes)
summary(fimllm, Nbt = 5)
```

Index

*** Full Information Maximum Likelihood**

 fimlreg, 5

*** Linear Model**

 fimlreg, 5

 ilse, 6

*** Missing Value**

 ilse, 6

Coef, 2

cor.mat, 3

cov.mat, 4

fimlreg, 5

Fitted.values (summary), 11

ilse, 6, 6

kern, 8

lm, 8

nhanes, 9

print, 10

Residuals (summary), 11

summary, 11