Package ‘plmm’

February 20, 2015

Type Package

Title Partially Linear Mixed Effects Model

Version 0.1-1

Date 2012-11-28

Author OHINATA Ren

Maintainer OHINATA Ren <Ren.Ohinata@wiwi.uni-goettingen.de>

Depends sm, Formula, nlme

Description This package fits the partially linear mixed effects model (semiparametric random intercept model) using kernel regression, without distributional assumptions for the random terms. Estimation procedure is an iterative generalized least squares type. A nonparametric heteroskedastic variance function is allowed for the regression error. Bootstrap resampling is provided for inference. The package implements bandwidth selection by an alternative cross validation for correlated data.

License GPL (>= 2)

Repository CRAN

Date/Publication 2012-11-28 15:35:50

NeedsCompilation no

R topics documented:

`plmm-package` *Partially Linear Mixed Effects Model*

Description

This package fits the partially linear mixed effects model (semiparametric random intercept model) using kernel regression, without distributional assumptions for the random terms. Estimation procedure is an iterative generalized least squares type. A nonparametric heteroskedastic variance function is allowed for the regression error. Bootstrap resampling is provided for inference. The package implements bandwidth selection by an alternative cross validation for correlated data.

Details

Package:	plmm
Type:	Package
Version:	0.1-1
Date:	2012-11-28
License:	GPL (>= 2)

Author(s)

OHINATA Ren

Maintainer: OHINATA Ren <Ren.Ohinata@wiwi.uni-goettingen.de>

References

Ohinata R. (2012). Partially Linear Mixed Effects Model without Distributional Assumptions. Working Paper.

`coef` *Extract Fixed Regression Coefficients*

Description

Extract the fixed regression coefficient estimates of a model fitted using a model fitting function `plmm` or `wplmm`. `coefficients` is an alias.

Usage

```
## S3 method for class 'plmm'
coef(object,...)
## S3 method for class 'wplmm'
coef(object,...)
## S3 method for class 'plmm'
coefficients(object,...)
## S3 method for class 'wplmm'
coefficients(object,...)
```

Arguments

`object`	a model fitted with `plmm` or `wplmm`.
`...`	other arguments.

Details

When the `object` is of the ‘plmm’ class, the estimates from the final iteration are returned unless the model was fitted without iteration. The returned value `coef.iter` of a ‘plmm’ class object contains all the estimates computed in the iterations.

Value

Fixed regression coefficients estimates.

See Also

`plmm`, `wplmm`

Examples

```
data(plmm.data)
model <- plmm(y0~x1+x2+x3|t1, random=cluster, data=plmm.data)
coef(model)
model$coef.iter
```

`fitted` *Extract Model Fitted Values*

Description

Extract the predicted response values of a model fitted using a model fitting function `plmm` or `wplmm`. `fitted.values` is an alias.

Usage

```
## S3 method for class 'plmm'
fitted(object,...)
## S3 method for class 'wplmm'
fitted(object,...)
## S3 method for class 'plmm'
fitted.values(object,...)
## S3 method for class 'wplmm'
fitted.values(object,...)
```

Arguments

`object`	a model fitted with `plmm` or `wplmm`.
`...`	other arguments.

Details

The fitted value is defined as the conditional prediction, i.e. the sum of the estimated fixed components (parametric and nonparametric) and the predicted random intercept. Unconditional predictions (not conditioned on the cluster which the predicted response belongs to) can be obtained using `predict`.

Value

The predicted conditional response values.

See Also

`plmm`, `wplmm`, `predict`

Examples

```
data(plmm.data)
model <- plmm(y0~x1+x2+x3|t1, random=cluster, data=plmm.data)
fitted(model)
```

plmm *Partially Linear Mixed Effects Model*

Description

Fit a partially linear mixed effects model under homoskedastic regression error assumption. `plmm` returns an object of the ‘plmm’ class.

Usage

```
plmm(formula, random, h0, data, vc.method = "FC", nonpar.bws = "h.select",
poly.index = 1, iter = 20, scale.h = 1, epsilon = 0.003, lim.binning = 100,
hetero.prop = NULL, ...)
```

Arguments

`formula` a symbolic description of the model to be fitted under homoskedastic regression error assumption. `formula` consists of three parts: the response (the left hand side of ‘~’), the fixed parametric component (between ‘~’ and ‘|’), and the fixed nonparametric component (the right hand side of ‘|’).

`random` the name of the clustering factor variable. The levels of the factor can be either numeric or character.

`h0` a set of bandwidths for nonparametric kernel regression to reduce the model. `h0` can be obtained using `select.h0`. `h0` is optional; if omitted, `select.h0` is called automatically to compute a set of bandwidths. The user can modify bandwidths in a list object created by `select.h0` and pass the object to `plmm`.

`data` an optional data frame containing the variables in the model. If relevant variables are not found in `data`, the variables are taken from the environment `plmm` was called from.

`vc.method` the method for variance components estimation: either the default “FC” (fitting of constant) or “SA” (Swamy-Arora). If heteroskedasticity proportionalities are known, “FChetero” (heteroskedastic fitting of constant) and “SAhetero” (heteroskedastic Swamy-Arora) can be used with `hetero.prop` specified.

`nonpar.bws` the bandwidth selection method for the kernel regression of the nonparametric component. “h.select” (default), “hcv”, “GCV” and “GCV.c” are available.

`poly.index` the degree of polynomial for the kernel regression of the nonparametric component: either 0 for local constant or 1 (default) for local linear.

`iter` the maximum number of iterations to be performed. The default is 20.

`scale.h` a scalar or 2-dimensional vector to scale the bandwidths selected for kernel regression of the nonparametric component. The default is 1. When a scalar is given for a nonparametric component of two covariates, it scales the bandwidths in both directions by the same factor.

`epsilon` The criterion to determine the convergence of model fitting. The default value is 0.003.

`lim.binning` the smallest sample size below which binning techniques are not used to calculate the degrees of freedom of the estimated nonparametric component (the ordinary cross-validation is used instead). This option doesn't apply if "GCV.c" is used for `nonpar.bws`.

`hetero.prop` a vector of heteroskedasticity proportionalities. The length of the vector should be the same as either the sample size (within-cluster heteroskedasticity) or the number of the clusters (between-cluster heteroskedasticity).

`...` optional arguments relevant to `h.select` or `hcv`, which include `nbins`, `hstart` and `hend`. See `sm.options` and `hcv`.

Details

The number of covariates of the nonparametric component is at most two. Kernel regression is estimated using `sm.regression`. There are four methods for bandwidth selection: "h.select" calls `h.select` to execute cross validation (CV) using binning techniques; "hcv" calls `hcv` which implements the ordinary CV; "GCV" uses the generalized CV; and "GCV.c" performs generalized CV for correlated data. `sm.regression`, `h.select` and `hcv` are functions of the **sm** package. When the nonparametric component is a function of two variables, optimization procedure selects one bandwidth that, multiplied by the standard deviations of those variables, minimizes the cross validation statistic. The user can further scale the bandwidths using `scale.h`. `epsilon` is the value to determine the convergence of iterative estimation. For the rth iteration round, the absolute value of $(\sigma^2_{(r)} - \sigma^2_{(r-1)})/\sigma^2_{(r-1)}$ is calculated for each variance component. The iteration procedure ends when this absolute value of both variance components becomes smaller than `epsilon`.

Value

`coefficients` estimated regression coefficients.

`fitted.values` conditional predictions of the response, defined as the sum of the estimated fixed components and the predicted random intercepts.

`residuals` residuals of the fitted model, defined as the response values minus the conditional predictions of the response.

`var.comp` variance component estimates.

`nonpar.values` estimated function values of the nonparametric component at the data points.

`h.nonpar` the bandwidths used to estimate the nonparametric component in the final iteration.

`rank` the degrees of freedom of the parametric component, which doesn't include the intercept term.

`df.residual` the residual degrees of freedom defined as $N - p - tr(2S - S^T)$ where N is the sample size, p is the rank of the parametric component, and S is the smoother matrix for the nonparametric component. If "GCV.c" is used for `nonpar.bws`, an alternative definition $N - p - tr(2SR - SRS^T)$ is applied with R being the estimated correlation matrix of the data.

`nbins` the number of bins (which would have been) used for binning for cross validation and the calculation of the degrees of freedom.

`iter` the number of iterations performed.

`coef.iter` regression coefficient estimates computed in the iteration process.

`vc.iter` variance component estimates computed in the iteration process.

`formula` formula passed to `plmm`.

`call` the matched call to `plmm`.

`h0.call` the matched call to `select.h0`.

`xlevels` if there are factors among the covariates in the parametric component, the levels of those factors.

See Also

`select.h0`, `h.select`, `hcv`, `sm.options`.

Examples

```
data(plmm.data)
plmm(y0~x1+x2+x3|t1, random=cluster, data=plmm.data)

# heteroskedasticity proportionality x3
plmm(y1~x1+x2+x3|t1, random=cluster, data=plmm.data, vc.method="FChetero", hetero.prop=x3)

# nonparametric component of two covariates, t1 and t2
## Not run:
plmm(y2~x1+x2+x3|t1+t2, random=cluster, data=plmm.data)
## End(Not run)
```

`plmm.bs` *Bootstrap Inference*

Description

The fixed regression coefficients and the random effects variance are repeatedly estimated using the wild bootstrap to estimate the sampling distribution of the estimators.

Usage

```
plmm.bs(object, B, data, h0, ...)
```

Arguments

`object` a model estimated using a model fitting function `plmm` or `wplmm`.

`B` the number of bootstrap replications.

`data` an optional data frame containing the variables in the model. If relevant variables are not found in `data`, the variables are taken from the environment from which `plmm.bs` was called.

`h0` a set of bandwidths for nonparametric kernel regression to reduce the model. `h0` is optional; if omitted, `select.h0` is called automatically to compute bandwidths according to the `h0.call` value of the `object`.

`...` other arguments.

Details

plmm.bs computes bootstrap estimates of the fixed regression coefficients and the random effects variance using the wild bootstrap resampling technique. The user can modify bandwidths in a list object created by select.h0 and pass the object to plmm.bs. In the wild bootstrap procedure, only the bandwidth for the kernel estimation of the response is recalculated while those for the fixed parametric component remain the same as given in h0. plmm.bs returns an object of bs.plmm class for which the summary method is available.

Value

A list of a matrix of B rows. Its columns contain B bootstrap estimates of the fixed regression coefficients and random effects variance.

See Also

summary.bs.plmm, select.h0.

Examples

```
data(plmm.data)
h0 <- select.h0(y1~x1+x2+x3|t1, data=plmm.data)
model <- plmm(y1~x1+x2+x3|t1, h0=h0, random=cluster, data=plmm.data)
model2 <- wplmm(model, heteroX=x3, data=plmm.data)

## Not run:
bs <- plmm.bs(model, B=500, data=plmm.data, h0=h0)
bs2 <- plmm.bs(model2, B=500, data=plmm.data, h0=h0)
## End(Not run)
```

plmm.data *Data Set for the Package Examples*

Description

plmm.data is a data.frame of simulated, hierarchically structured (2-level) data used in examples provided in the package documents.

Usage

```
data(plmm.data)
```

Format

A data frame with 363 observations on the following 9 variables.

y0 a numeric vector of the response simulated with one covariate for the nonparametric component under homoskedastic regression error assumption.

y1 a numeric vector of the response simulated with one covariate for the nonparametric component under heteroskedastic regression error assumption.

y2 a numeric vector the response simulated with two covariates for the nonparametric component under homoskedastic regression error assumption.

x1 a factor with levels `{F, M}`.

x2 a numeric vector of a covariate in the fixed parametric component.

x3 a numeric vector of a covariate in the fixed parametric component.

t1 a numeric vector of a covariate in the fixed nonparametric component.

t2 a numeric vector of a covariate in the fixed nonparametric component.

cluster a clustering variable. There are 20 clusters. The cluster size is unbalanced; each cluster consists of from 5 to 30 observations.

Details

Data are generated from the following three models:

$$y0 = \beta_0 + \beta_1 x1 + \beta_2 x2 + \beta_3 x3 + \gamma_1(t1) + u + e_0$$

$$y1 = \beta_0 + \beta_1 x1 + \beta_2 x2 + \beta_3 x3 + \gamma_1(t1) + u + e_1$$

$$y2 = \beta_0 + \beta_1 x1 + \beta_2 x2 + \beta_3 x3 + \gamma_2(t1, t2) + u + e_1$$

where

$$(\beta_0, \beta_1, \beta_2, \beta_3) = (1, 1, 0.5, -1)$$

$$\gamma_1(t1) = 2sin(t1)$$

$$\gamma_2(t1, t2) = 2sin(t1) + 0.3t2 + 0.75t2^2$$

$$u \sim N(0, 3)$$

$$e_0 \sim N(0, 1)$$

$$e_1 = x3 * e_0$$

u is the cluster-specific random intercept. x2 and t1 are correlated with correlation coefficient about 0.67.

Examples

```
data(plmm.data)
str(plmm.data)
```

plot *Nonparametric Function Plot*

Description

Create a two or three dimensional plot of the estimated nonparametric function.

Usage

```
## S3 method for class 'plmm'
plot(x, data, ...)
## S3 method for class 'wplmm'
plot(x, data, ...)
```

Arguments

`x`	a model fitted using a model fitting function `plmm` or `wplmm`.
`data`	an optional data frame containing the variables in the model. If relevant variables are not found in data, the variables are taken from the environment `plmm` was called from.
`...`	optional arguments relevant to creating a plot using `sm.regression`.

Details

Along the vertical axis is the value obtained by subtracting the estimated fixed parametric component from the response. A two or three dimensional plot is created using `sm.regression`. Some of the relevant optional arguments include `display`, `nbins` and `ngrid`. See `sm.options`.

See Also

`sm.regression`, `sm.options`

Examples

```
data(plmm.data)
model <- plmm(y0~x1+x2+x3|t1, random=cluster, data=plmm.data)
plot(model, data=plmm.data, ann="F")
title(ylab=expression(paste("y-X",hat(beta))))

## Not run:
model <- plmm(y2~x1+x2+x3|t1+t2, random=cluster, data=plmm.data)
plot(model, data=plmm.data, display="rgl")
## End(Not run)
```

predict — *Conditional or Unconditional Model Prediction*

Description

Predict the response values for a set of covariates based on a model fitted using a model fitting function `plmm` or `wplmm`.

Usage

```
## S3 method for class 'plmm'
predict(object, newdata, data, cond = TRUE, ...)
## S3 method for class 'wplmm'
predict(object, newdata, data, cond = TRUE, ...)
```

Arguments

`object`	a model fitted with `plmm` or `wplmm`.
`newdata`	a data frame containing sets of values of the covariates. The clustering variable may be omitted, in which case only unconditional prediction is possible. If `newdata` is omitted, the response of the original data set is predicted conditionally or unconditionally according to cond.
`data`	an optional data frame containing the variables used to fit the model. If relevant variables are not found in `data`, the variables are taken from the environment from which `predict` was called.
`cond`	logical. If `TRUE`, the response is predicted conditional on the random effect prediction for the cluster the observational unit belongs to. If `FALSE`, an unconditional prediction of the response, i.e. the sum of only the fixed components is computed.
`...`	other arguments.

Value

Predicted conditional or unconditional response values.

Examples

```
data(plmm.data)
model <- plmm(y0~x1+x2+x3|t1, random=cluster, data=plmm.data)
pred_data <- data.frame(x1=c("F","M"), x2=c(15,25), x3=c(3,2), t1=c(5, 4), cluster=c(1,5))
predict(model, newdata=pred_data, data=plmm.data)
predict(model, newdata=pred_data, data=plmm.data, cond=FALSE)
```

ranef *Random Effects Prediction*

Description

Predict the random effects (intercepts) of a model fitted using a model fitting function plmm or wplmm.

Usage

```
## S3 method for class 'plmm'
ranef(object, data, ...)
## S3 method for class 'wplmm'
ranef(object, data, ...)
```

Arguments

object	a model fitted with plmm or wplmm.
data	an optional data frame containing the variables in the model. If relevant variables are not found in data, the variables are taken from the environment plmm was called from.
...	other arguments.

Details

The predictor is linear in the response. A random effect is predicted for each element of the clustering variable specified for random in the function call to plmm.

Value

A named vector of predicted random effects.

Examples

```
data(plmm.data)
model <- plmm(y0~x1+x2+x3|t1, random=cluster, data=plmm.data)
ranef(model, data=plmm.data)
```

residuals *Extract Model Residuals*

Description

Extract residuals from a model fitted using a model fitting function plmm or wplmm. resid is an alias.

Usage

```
## S3 method for class 'plmm'
residuals(object,...)
## S3 method for class 'wplmm'
residuals(object,...)
## S3 method for class 'plmm'
resid(object,...)
## S3 method for class 'wplmm'
resid(object,...)
```

Arguments

object	a model fitted with plmm or wplmm.
...	other arguments.

Details

The residual is defined as the value obtained by subtracting from the response value the estimated fixed components and the predicted random intercept of the cluster the observational unit of the response belongs to.

See Also

plmm, wplmm

Examples

```
data(plmm.data)
model <- plmm(y0~x1+x2+x3|t1, random=cluster, data=plmm.data)
residuals(model)
```

select.h0 *Bandwidths Selection for Model Reduction*

Description

Select the bandwidths for kernel regression to reduce the partially linear mixed effects model to a mixed effects model.

Usage

```
select.h0(formula, data, nonpar.bws = "h.select", poly.index = 1, ...)
```

Arguments

formula
: a symbolic description of the model to fit with the model fitting function plmm. formula consists of three parts: the response (the left hand side of '~'), the fixed parametric component (between '~' and '|'), and the fixed nonparametric component (the right hand side of '|').

data
: an optional data frame containing the variables in the formula. If relevant variables are not found in data, they are taken from the environment from which select_h0 was called.

nonpar.bws
: the cross validation method for bandwidth selection. The method is either the default "h.select" (cross validation using binning technique) or "hcv" (ordinary cross validation).

poly.index
: the degree of polynomial of the kernel regression: either 0 for local constant or 1 (default) for local linear.

...
: optional arguments relevant to h.select or hcv, which include nbins, hstart and hend. See sm.options and hcv.

Details

select.h0 yields a list object that can be used for the argument h0 in the model fitting function plmm. Bandwidths are selected for kernel regression of the response and the covariates in the fixed parametric component. "h.select" uses binning techniques for cross validation. The number of bins for binning is set to the default integer, the rounded value of 8*log(N)/d), where N is the sample size and d is the number of covariates in the nonparametric component. When the sample size is small (N < 100), binning techniques are not used and the bandwidths selected will coincide with those obtained with "hcv".

Value

h0
: a vector (if the nonparametric component is a function of one variable) or a matrix (if it is a function of two variables) of bandwidths selected.

nbins
: the number of bins (which would be) used for binning.

h0.call
: the matched call to select.h0.

See Also

plmm, h.select, hcv.

Examples

```
data(plmm.data)
select.h0(y0~x1+x2+x3|t1, data=plmm.data)

# nonparametric component of two covariates, t1 and t2
## Not run:
select.h0(y2~x1+x2+x3|t1+t2, data=plmm.data)
## End(Not run)
```

summary *Summary of a Fitted Partially Linear Mixed Effects Model.*

Description

Summarize a partially linear mixed effects model fitted using a model fitting function `plmm` or `wplmm`.

Usage

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

Arguments

`object`	a model fitted with `plmm` or `wplmm`.
`...`	other arguments.

Value

`coefficients`	estimated regression coefficients.
`var.comp`	estimated variance components.
`h.nonpar`	the bandwidths used to estimate the nonparametric component in the final iteration.
`var.fun.h`	(only if `object` is of the ‘wplmm’ class) the bandwidths used to estimate the variance function.
`iter`	(only if object is of the ‘plmm’ class) the number of iterations.
`formula`	formula passed to `plmm`.
`call`	the matched call to `plmm` or `wplmm` that returned the `object`.

plmm.call	(only if object is of the 'wplmm' class) the matched call to plmm that returned the object.
heteroX	(only if object is of the 'wplmm' class) the names of the heteroskedasticity conditioning variables.

See Also

plmm, wplmm

Examples

```
data(plmm.data)
model <- plmm(y0~x1+x2+x3|t1, random=cluster, data=plmm.data)
summary(model)
```

summary.bs.plmm *Summary of the bootstrap estimates of the sampling distribution.*

Description

Provide summary statistics of the estimated sampling distributions of the fixed regression coefficient estimators and the random effects variance estimator.

Usage

```
## S3 method for class 'bs.plmm'
summary(object, probs = c(0.005, 0.025, 0.05, 0.95, 0.975, 0.995), ...)
```

Arguments

object	an object returned from plmm.bs.
probs	a vector of quantiles.
...	other arguments.

Details

For an object of the bs.plmm class, summary provides the mean, the standard deviation and quantiles of the estimated marginal sampling distributions of the coefficient estimators and the random effects variance estimator.

Value

mean	a vector of the means.
sd	a vector of the standard deviations.
quantiles	vectors of the quantiles specified for probs.

Examples

```
data(plmm.data)
h0 <- select.h0(y0~x1+x2+x3|t1, data=plmm.data)
model <- plmm(y0~x1+x2+x3|t1, h0=h0, random=cluster, data=plmm.data)
## Not run:
bs <- plmm.bs(model, B=500, data=plmm.data, h0=h0)
summary(bs, probs=c(0.025, 0.975))
## End(Not run)
```

`var.plot` *Nonparametric Variance Function Estimation and Plotting*

Description

Plot the estimated nonparametric variance function and provide the estimated function values.

Usage

```
var.plot(object, heteroX, data, var.fun.bws = "ROT", var.fun.poly.index = 0, ...)
```

Arguments

`object` a model fitted using the model fitting function `plmm`.

`heteroX` at most two variables conditioning the heteroskedasticity of the regression error variance. If there are two variables, they can be passed either as a 2-element list or a 2-column matrix.

`data` an optional data frame containing the variables in the model. If relevant variables are not found in `data`, the variables are taken from the environment from which `var.plot` was called.

`var.fun.bws` the bandwidth selection method for the kernel regression estimation of the variance function. A rule-of-thumb type method "ROT" (default), "h.select" (cross validation using binning technique) or "hcv" (ordinary cross validation are available.

`var.fun.poly.index` the degree of polynomial of the kernel regression to estimate the nonparametric variance function: either 1 for local linear or 0 (default) for local constant.

`...` optional arguments relevant to estimation and plotting with `sm.regression`.

Details

The variance function plotted is an unconditional estimate, i.e. the sum of the estimated variances of the random effects and the regression error. As opposed to `wplmm`, `var.plot` does not trim negative estimates of the variance function values. `var.fun.values` returned from `var.plot` are also untrimmed estimates. "ROT" selects the bandwidths for heteroskedasticity conditioning variable w by $sd(w)N^{-1/(4+q)}$ where q is th the number of the conditioning variables (1 or 2) and N is the sample size. Some of the relevant optional arguments include `display`, `nbins` and `ngrid`. See `sm.options`.

Value

The following values are returned invisibly (they are not printed, but can be assigned).

`var.fun.values`	the estimated untrimmed conditional variance function values at the data points.
`var.comp`	the estimated variance of the random effects.
`h.var.fun`	the bandwidths used to estimate the nonparametric variance function.

See Also

`wplmm`, `sm.regression`, `sm.options`

Examples

```
data(plmm.data)
model <- plmm(y1~x1+x2+x3|t1, random=cluster, data=plmm.data)
var.plot(model, heteroX=x3, data=plmm.data)
result <- var.plot(model, heteroX=x3, data=plmm.data, display="none")
result$var.fun.values
```

---

wplmm *Weighted Partially Linear Mixed Effects Model*

Description

Estimate the regression error variance function nonparametrically from a partially linear mixed effects model fitted using the model fitting function `plmm`, and refit the model applying the weighted least squares procedure. `wplmm` returns an object of the 'wplmm' class.

Usage

```
wplmm(object, heteroX, data, nonpar.bws = "h.select", poly.index = 1,
var.fun.bws = "ROT", var.fun.poly.index = 0, scale.h = 1, trim = 0.01,
lim.binning = 100, ...)
```

Arguments

`object` a model fitted with `plmm`.

`heteroX` at most two variables conditioning the heteroskedasticity of the regression error variance. If there are two variables, they can be passed either as a 2-element list or a 2-column matrix.

`data` an optional data frame containing the variables in the model. If relevant variables are not found in data, they are taken from the environment from which `wplmm` was called.

`nonpar.bws` the bandwidth selection method for the kernel regression of the nonparametric component. The default method "h.select" (cross validation (CV) using binning technique), "hcv" (ordinary CV), "GCV" (generalized CV) and "GCV.c" (generalized CV for correlated data) are available.

`poly.index` the degrees of polynomial for the kernel regression of the nonparametric component: either 0 for local constant or 1 (default) for local linear.

`var.fun.bws` the bandwidth selection method for kernel regression of the variance function. A rule-of-thumb type method “ROT” (default), “h.select” (cross validation using binning technique) and “hcv” (ordinary cross validation) are available.

`var.fun.poly.index`
the degree of polynomial of the kernel regression to estimate the nonparametric variance function: either 0 (default) for local constant or 1 for local linear.

`scale.h` a scalar or 2-dimensional vector to scale the bandwidths selected for kernel regression of the nonparametric component. The default is 1. When a scalar is given for a nonparametric component of two covariates, it scales the bandwidths in both directions by the same factor.

`trim` if estimated variance function values are below the value of `trim`, they are set to this value. The default is 0.01.

`lim.binning` the smallest sample size below which binning techniques are not used to calculate the degrees of freedom of the estimated nonparametric component. Then, the ordinary cross-validation is used instead. This option doesn’t apply if “GCV.c” is used for `nonpar.bws`.

`...` optional arguments relevant to `h.select` or `hcv`, which include `nbins`, `hstart` and `hend`. See `sm.options` and `hcv`.

Details

There are three methods to select bandwidths for kernel regression of the nonparametric variance function: “h.select” and “hcv” call `h.select` and `hcv`, respectively, which are functions of the **sm** package; “ROT” selects the bandwidths for heteroskedasticity conditioning variable w by $sd(w)N^{-1/(4+q)}$ where q is the number of the conditioning variables (1 or 2) and N is the sample size.

Value

`coefficients` estimated regression coefficients.

`fitted.values` conditional predictions of the response, defined as the sum of the estimated fixed components and the predicted random intercepts.

`residuals` residuals of the fitted model, defined as the response values minus the conditional predictions of the response.

`var.comp` variance component estimates.

`nonpar.values` estimated function values of the nonparametric component at the data points.

`h.nonpar` the bandwidths used to estimate the nonparametric component.

`var.fun.values` estimated variance function values. Original computations less than the value of `trim` have been set to the value of `trim`.

`h.var.fun` the bandwidths used to estimate the nonparametric variance function.

`rank` the degrees of freedom of the parametric component, which doesn’t include the intercept term.

`df.residual` the residual degrees of freedom defined as $N - p - tr(2S - S^T)$ where N is the sample size, p is the rank of the parametric component, and S is the smoother matrix for the nonparametric component. If "GCV.c" is used for `nonpar.bws`, alternative definition $N - p - tr(2SR - SRS^T)$ is applied with R being the estimated correlation matrix of the data.

`nbins` the number of bins (which would have been) used for binning for CV and the calculation of the degrees of freedom.

`formula` formula passed to `wplmm`.

`call` the matched call to `wplmm`.

`h0.call` the matched call to `select.h0` underlying the `plmm` that yielded the `object`.

`plmm.call` the matched call to the `plmm` that yielded the `object`.

`xlevels` if there are factors among the covariates in the parametric component, the levels of those factors.

`heteroX` the names of the heteroskedasticity conditioning variables.

See Also

`plmm`, `h.select`, `hcv`, `sm.options`.

Examples

```
data(plmm.data)
model <- plmm(y1~x1+x2+x3|t1, random=cluster, data=plmm.data)
model2 <- wplmm(model, heteroX=x3, data=plmm.data)
summary(model2)
```

Index