Package ‘REEMtree’

February 19, 2015

Type Package

Title Regression Trees with Random Effects for Longitudinal (Panel) Data

Version 0.90.3

Date 2011-07-15

Author Rebecca Sela and Jeffrey Simonoff

Maintainer Rebecca Sela <rsela@stern.nyu.edu>

Depends nlme, rpart, methods

Suggests AER

Description This package estimates regression trees with random effects as a way to use data mining techniques to describe longitudinal or panel data.

License GPL

URL http://pages.stern.nyu.edu/~jsimonof/REEMtree/

Repository CRAN

Date/Publication 2011-08-08 05:41:32

NeedsCompilation no

R topics documented:

REEMtree-package .................................................. 2
AutoCorrelationLRtest .......................................... 2
fitted.REEMtree ................................................... 4
is ................................................................. 5
logLik ............................................................. 6
plot ................................................................. 7
predict ............................................................. 8
print ............................................................... 9
ranef.REEMtree ................................................ 10
REEMtree .......................................................... 11
REEMtree.object .................................................. 13
**Description**

This package estimates regression trees with random effects as a way to use data mining techniques to describe longitudinal or panel data.

**Details**

- **Package:** REEMtree
- **Type:** Package
- **Version:** 1.0
- **Date:** 2009-05-07
- **License:** GPL

**Author(s)**

Rebecca Sela <rsela@stern.nyu.edu>

**References**


**Examples**

```r
data(simplereemdata)
REEMresult <- REEMtree(Y~D+t+X, data=simplereemdata, random=~1|ID)
print(REEMresult)
```

**AutoCorrelationLRtest**  
*Test for autocorrelation in the residuals of a RE-EM tree*
Description

This function tests for autocorrelation in the residuals of a RE-EM tree using a likelihood ratio test. The test keeps the tree structure of the RE-EM tree object fixed and uses a standard likelihood ratio test on the linear random effects model.

Usage

AutoCorrelationLRtest(object, newdata=NULL, correlation=corAR1())

Arguments

- object: A RE-EM tree
- newdata: Dataset on which the test is to be performed; if none is given, the original dataset is used
- correlation: Type of correlation to be tested for in the residuals. The correlation can be any of type corClasses.

Details

In general, newdata is likely to be the data used to estimate object. The RE-EM tree can be estimated with or without allowing for autocorrelation. Because the estimated tree may differ depending on whether autocorrelation is allowed in the RE-EM tree estimation process, but we recommend testing based on the tree estimated with autocorrelation allowed and the tree estimated without autocorrelation allowed.

Value

- correlation: Type of correlation used in testing
- loglik0: Likelihood of the random effects model if there is no autocorrelation
- loglikAR: Likelihood of the random effects model if autocorrelation (of type AR(1)) is estimated
- pvalue: P-value of the likelihood ratio test

Author(s)

Rebecca Sela <rsela@stern.nyu.edu>

References


See Also

corClasses
Examples

data(simplereemdata)

# Estimation without autocorrelation
simpleEMresult<-REEMtree(Y=D+t+X, data=simplereemdata, random=-1|ID)
# Estimation with autocorrelation
simpleEMresult2<REEMtree(Y=D+t+X, data=simplereemdata, random=-1|ID, correlation=corAR1())

# Autocorrelation test based on the first tree
AutoCorrelationLRtest(simpleEMresult, simplereemdata)
# Autocorrelation test based on the second tree
AutoCorrelationLRtest(simpleEMresult2, simplereemdata)
# Autocorrelation test with an alternative correlation structure
AutoCorrelationLRtest(simpleEMresult, simplereemdata, correlation=corCAR1())


---

fitted.REEMtree

*Extract the fitted values from a RE-EM tree*

Description

This function extracts the fitted values from the LME object underlying the RE-EM tree. The fitted values are the fixed effects (from the tree) plus the estimated contributions of the random effects to the fitted values at grouping levels less or equal to the level given.

Usage

fitted.REEMtree(object, level, asList, ...)

Arguments

- **object**: an object of class REEMtree
- **level**: the level of random effects used in creating fitted values. Level 0 is fixed effects; levels increase with the grouping of random effects. Default is the highest level.
- **asList**: an optional logical value. If TRUE and a single value is given in level, the returned object is a list with the fitted values split by groups; otherwise the returned value is either a vector or a data frame, according to the length of level. Defaults to FALSE.
- **...**: some methods for this generic require additional arguments; none are used here.

Value

If the level is a single value, the result is a vector or list (depending on asList) with the fitted values. Otherwise, the result is a data frame with columns given by the fitted values at different levels.
Author(s)
Rebecca Sela <rsela@stern.nyu.edu>

References

See Also
fitted, REEMtree.object

Examples

data(simpleREEMdata)
REEMresult <- REEMtree(Y=D+t+X, data=simpleREEMdata, random=-1|ID)
fitted(REEMresult)

is
Is a RE-EM tree object

Description
This function tests whether an object is of the REEMtree class.

Usage
is.REEMtree(object)

Arguments
object any R object

Value
TRUE if the object is of the REEMtree type

Author(s)
Rebecca Sela <rsela@stern.nyu.edu>

References
Examples

```
data(simplereemdata)
REEMresult<-REEMtree(Y~D+t+X, data=simplereemdata, random=-1|ID)
is.REEMtree(REEMresult)
```

---

**logLik**

Log-likelihood of a RE-EM tree

**Description**

This returns the log-likelihood of the effects model of a RE-EM tree. This is the log-likelihood of the random effects model estimated in the RE-EM tree. (The regression tree is not associated with a log-likelihood.)

**Usage**

```
logLik.REEMtree(object,...)
```

**Arguments**

- `object`: an object of class REEMtree
- `...`: further arguments passed to or from other methods

**Value**

the log-likelihood of the fitted effects model associated with \( x \)

**Author(s)**

Rebecca Sela <rsela@stern.nyu.edu>

**References**


**See Also**

`REEMtree.object`

**Examples**

```
data(simplereemdata)
REEMresult<-REEMtree(Y~D+t+X, data=simplereemdata, random=-1|ID)
logLik(REEMresult)
```
Description

Plots the regression tree associated with a RE-EM tree.

Usage

`plotREEmtree(x, text = TRUE, ...)`

Arguments

- `x`: a fitted object of class REEmtree
- `text`: if TRUE, the text of the tree will be plotted on the tree automatically.
- `...`: further arguments passed to or from other methods

Value

The coordinates of the nodes are returned as a list, with components `x` and `y`.

Author(s)

Rebecca Sela <rsela@stern.nyu.edu>

References


See Also

`REEmtree, plot.rpart`

Examples

data(simpleREEmdata)
REEmresult<-REEmtree(Y~D+t+X, data=simpleREEmdata, random=-1|ID)
plot(REEmresult)
predict

Predictions from a regression tree with individual-specific effects

Description

Returns a vector of predictions from a fitted RE-EM Tree. Predictions are based on the node of the
tree in which the new observation would fall and (optionally) an estimated random effect for the
observation.

Usage

predict.REEMtree(object, newdata, id = NULL,
EstimateRandomEffects = TRUE, ...)

Arguments

- **object**: a fitted REEMtree
- **newdata**: an data frame to be used for obtaining the predictions. All variables used in the
fixed and random effects models, including the group identifier, must be present
in the data frame. New values of the group identifier are allowed. Unlike in
predict.lme and predict.rpart, the data frame is required
- **id**: a string containing the name of the variable that is used to identify the groups.
This is required if EstimateRandomEffects=TRUE and newdata does not match
the data used to estimate the random effects model that created object.
- **EstimateRandomEffects**: if TRUE, the fitted effects will be included in the estimates and effects for new
groups will be estimated wherever the target variable is not missing. If FALSE
or if the random effect cannot be estimated, random effects are set to 0, so that
only the fixed effects based on the regression tree are used.
- **...**: additional arguments that will be passed through to rpart

Details

If EstimateRandomEffects=TRUE and a group was not used in the original estimation, its random
effect must be estimated. If there are no non-missing values of the target variable for this group,
then the new effect is set to 0.

If there are non-missing values of the target variable, then the random effect is estimated based on
the estimated variance of the errors and variance of the random effects in the fitted model. See
Equation 3.2 of Laird and Ware (1982) for the precise relationship.

Important note: In this implementation, estimation of group effects for new groups can be used only
with group-specific intercepts are estimated with only one grouping variable.

Value

a vector containing the predicted values
Author(s)

Rebecca Sela <rsela@stern.nyu.edu>

References


See Also

`predict.nlme`, `predict.rpart`

Examples

data(simplereemdata)
REEMresult <- REEMtree(Y~D+t+X, data=simplereemdata, random=~1|ID)
predict(REEMresult, simplereemdata, EstimateRandomEffects=FALSE)
predict(REEMresult, simplereemdata, id=simplereemdata$ID, EstimateRandomEffects=TRUE)

# Estimation based on a subset that excludes the last two time series, # with predictions for all observations
sub <- rep(c(rep(TRUE, 10), rep(FALSE, 2)), 50)
REEMresult <- REEMtree(Y~D+t+X, data=simplereemdata, random=~1|ID, subset=sub)
pred1 <- predict(REEMresult, simplereemdata, EstimateRandomEffects=FALSE)
pred2 <- predict(REEMresult, simplereemdata, id=simplereemdata$ID, EstimateRandomEffects=TRUE)

# Estimation based on a subset that excludes the last five individuals, # with predictions for all observations
sub <- c(rep(TRUE, 50), rep(FALSE, 60))
REEMresult <- REEMtree(Y~D+t+X, data=simplereemdata, random=~1|ID, subset=sub)
pred3 <- predict(REEMresult, simplereemdata, EstimateRandomEffects=FALSE)
pred4 <- predict(REEMresult, simplereemdata, id=simplereemdata$ID, EstimateRandomEffects=TRUE)

print

Print a RE-EM Tree object

Description

This function prints a description of a fitted RE-EM tree object.

Usage

`print.REEMtree(x,...)`
Arguments

x  fitted model of class REEMtree

... further arguments passed to or from other methods

Details

This function is a method for the generic function print for class REEMtree. It can be invoked by calling print for an object of class REEMtree, or by calling print.REEMtree directly for an object of the corresponding type.

Side Effects

Prints representations of the regression tree and the random effects model that comprise a RE-EM tree.

Author(s)

Rebecca Sela <rsela@stern.nyu.edu>

References


See Also

print.rpart, REEMtree.object

Examples

data(simpleREEMdata)
REEMresult<-REEMtree(Y~D+X, data=simpleREEMdata, random=-1|ID)
print(REEMresult)

---

**ranef.REEMtree**  
*Extract the estimated random effects from a RE-EM tree*

Description

This function extracts the estimated random effects from a fitted RE-EM tree.

Usage

ranef.REEMtree(object,...)
**Arguments**

- `object`: an object of class `REEMtree`
- `...`: further arguments passed to or from other methods

**Value**

A vector containing the estimated random effects

**Author(s)**

Rebecca Sela <rsela@stern.nyu.edu>

**References**


**See Also**

- `random.effects`, `REEMtree.object`

**Examples**

```r
data(simplereemdata)
reemresult <- REEMtree(Y~D+t+X, data=simplereemdata, random=-1|ID)
ranef(reemresult)
```

---

**Description**

Fit a RE-EM tree to data. This estimates a regression tree combined with a linear random effects model.

**Usage**

```r
REEMtree(formula, data, random, subset=NULL, initialRandomEffects=rep(0, TotalObs),
ErrorTolerance=0.001, MaxIterations=1000, verbose=FALSE, tree.control=rpart.control(),
cv=TRUE, cpmin = 0.001, no.SE =1,
lme.control=lmeControl(returnObject=TRUE), method="REML", correlation=NULL)
```
Arguments

formula  a formula, as in the \texttt{lm} or \texttt{rpart} function

data     a data frame in which to interpret the variables named in the formula (unlike in \texttt{lm} or \texttt{rpart}, this is not optional)

random   a description of the random effects, as a formula of the form \texttt{~1|g}, where \texttt{g} is the grouping variable

subset   an optional logical vector indicating the subset of the rows of data that should be used in the fit. All observations are included by default.

\texttt{initialRandomEffects}  an optional vector giving initial values for the random effects to use in estimation

\texttt{ErrorTolerance}  when the difference in the likelihoods of the linear models of two consecutive iterations is less than this value, the RE-EM tree has converged

\texttt{MaxIterations}  maximum number of iterations allowed in estimation

\texttt{verbose}  if \texttt{TRUE}, the current estimate of the RE-EM tree will be printed after each iteration

\texttt{tree.control}  a list of control values for the estimation algorithm to replace the default values used to control the \texttt{rpart} algorithm. Defaults to an empty list.

\texttt{cv}  if \texttt{TRUE} then cross-validation will be used for estimating the tree at each iteration. Default is \texttt{TRUE}.

\texttt{cpmin}  complexity parameter used in building a tree before cross-validation

\texttt{no.SE}  number of standard errors used in pruning (0 if unused)

\texttt{lme.control}  a list of control values for the estimation algorithm to replace the default values returned by the function \texttt{lmeControl}. Defaults to an empty list.

\texttt{method}  whether the linear model should be estimated with \texttt{ML} or \texttt{REML}

\texttt{correlation}  an optional \texttt{corStruct} object describing the within-group correlation structure; the available classes are given in \texttt{corClasses}

Value

an object of class \texttt{REEMtree}

Author(s)

Rebecca Sela <rsela@stern.nyu.edu>

References


See Also

\texttt{rpart}, \texttt{nlme}, \texttt{REEMtree.object}, \texttt{corClasses}
**Examples**

```r
data(simplereemdata)
REEMresult <- REEMtree(Y~D+t+X, data=simplereemdata, random=-1|ID)
```

# Estimation allowing for autocorrelation
```
REEMresult <- REEMtree(Y~D+t+X, data=simplereemdata, random=-1|ID, correlation=corAR1())
```

# Random parameters model for the random effects
```
REEMresult <- REEMtree(Y~D+t+X, data=simplereemdata, random=-1+X|ID)
```

# Estimation with a subset
```
sub <- rep(c(rep(TRUE, 10), rep(FALSE, 2)), 50)
REEMresult <- REEMtree(Y~D+t+X, data=simplereemdata, random=-1|ID, subset=sub)
```

# Dataset from the R library "AER"
```
data("Grunfeld", package = "AER")
REEMtree(invest ~ value + capital, data=Grunfeld, random=-1|firm)
REEMtree(invest ~ value + capital, data=Grunfeld, random=-1|firm, correlation=corAR1())
REEMtree(invest ~ value + capital, data=Grunfeld, random=-1+year|firm)
REEMtree(invest ~ value + capital, data=Grunfeld, random=-1|firm/year)
```

**Description**

Object representing a fitted REEMtree.

**Value**

- **Tree**: Fitted rpart tree associated with the fitted RE-EM tree
- **EffectModel**: fitted lme object associated with the fitted RE-EM tree
- **RandomEffects**: vector of estimated random effects
- **BetweenMatrix**: estimated variance of the random effects
- **ErrorVariance**: estimated variance of the errors
- **data**: the data frame used to estimate the RE-EM tree
- **logLik**: log likelihood of the linear model for the random effects
- **IterationsUsed**: number of iterations required to fit the REEMtree
- **Formula**: formula used in fitting the REEMtree
- **Random**: description of the random effects used in fitting the REEMtree
- **Groups**: the vector of group identifiers used in estimation
- **Subset**: the logical vector indicating the subset of the rows of data used in the fit
residuals.REEMtree

ErrorTolerance the error tolerance used in estimation

correlation the correlation structure used in fitting the linear model

residuals estimated residuals

method method (ML or REML) used in estimating the linear random effects model

lme.control parameters used to control fitting the linear random effects model

tree.control parameters used to control fitting the regression tree

Author(s)

Rebecca Sela <rsela@stern.nyu.edu>

References


See Also

*rpart, nlme, REEMtree*

Examples

```r
data(simplereemdata)
REEMresult <- REEMtree(Y~D+t+X, data=simplereemdata, random=-1|ID)
```

---

**residuals.REEMtree** Extract the residuals from a RE-EM tree

Description

This function extracts the residuals from the LME object underlying the RE-EM tree. The residuals depend on the fixed effects (from the tree) plus the estimated contributions of the random effects to the fitted values at grouping levels less or equal to the level given.

Usage

```r
residuals.REEMtree(object, level, type, asList, ...)
```

Arguments

- **object** an object of class REEMtree
- **level** the level of random effects used in creating residuals. Level 0 is fixed effects only; levels increase with the grouping of random effects. Default is the highest level.
**simpleREEMdata**

| type          | optional character string specifying the type of residuals to be used. If "response", the "raw" residuals (observed - fitted) are used. If "pearson", the standardized residuals (raw residuals divided by the corresponding standard errors) are used. If "normalized", the normalized residuals (standardized residuals premultiplied by the inverse square-root factor of the estimated error correlation matrix) are used. Only the first character needs to be provided. Defaults to "pearson".
| asList        | an optional logical value. If TRUE and a single value is given in level, the returned object is a list with the residuals split by groups; otherwise the returned value is either a vector or a data frame, according to the length of level. Defaults to FALSE.
| ...           | some methods for this generic require additional arguments; none are used here.

**Value**

If the level is a single value, the result is a vector or list (depending on asList) with the residuals. Otherwise, the result is a data frame with columns given by the residuals at different levels.

**Author(s)**

Rebecca Sela <rsela@stern.nyu.edu>

**References**


**See Also**

residuals, REEMtree.object

**Examples**

data(simpleREEMdata)
REEMresult<-REEMtree(Y~D+t+X, data=simpleREEMdata, random=~1|ID)
residuals(REEMresult)

---

**simpleREEMdata**

*Sample Data for RE-EM trees*

**Description**

This data set consists of a panel of 50 individuals with 12 observations per individual. The data is based on a regression tree with an initial split based on a dummy variable (D) and a second split based on time in the branch where D=1. The observations include both randomly generated individual-specific effects and observation-specific errors.
Format

The data has 600 rows and 5 columns. The columns are:

- `y` the target variable
- `t` a numeric predictor ("time")
- `d` a categorical predictor with two levels, 0 and 1
- `id` the identifier for each individual
- `x` another covariate (which is intentionally unrelated to the target variable)

References


---

**tree**

*Extract the regression tree associated with a RE-EM tree*

Description

Returns the fitted `rpart` object associated with a `REEMtree` object.

Usage

```r
tree(object,...)
```

Arguments

- `object` an object of class `REEMtree`
- `...` further arguments passed to or from other methods

Value

the fitted regression tree associated with the `REEMtree` object

Author(s)

Rebecca Sela <rsela@stern.nyu.edu>

References


See Also

`rpart.object`, `REEMtree.object`
Examples

data(simpleREEMdata)
REEMresult<-REEMtree(Y~D+t+X, data=simpleREEMdata, random=-1|ID)
tree.REEMtree(REEMresult)
tree(REEMresult)
Index

* Topic **classes**
  is, 5

* Topic **datasets**
  simpleREEMdata, 15

* Topic **hplot**
  plot, 7

* Topic **htest**
  AutoCorrelationLRtest, 2

* Topic **models**
  AutoCorrelationLRtest, 2
  fitted.REEMtree, 4
  is, 5
  logLik, 6
  predict, 8
  ranef.REEMtree, 10
  REEMtree, 11
  REEMtree.object, 13
  residuals.REEMtree, 14
  tree, 16

* Topic **print**
  print, 9

* Topic **tree**
  AutoCorrelationLRtest, 2
  is, 5
  logLik, 6
  plot, 7
  predict, 8
  print, 9
  REEMtree, 11
  REEMtree-package, 2
  REEMtree.object, 13
  tree, 16

AutoCorrelationLRtest, 2

corClasses, 3, 12

fitted, 5
fitted.REEMtree, 4

is, 5

logLik, 6

nlme, 12, 14

plot, 7
plot.rpart, 7
predict, 8
predict.nlme, 9
predict.rpart, 9
print, 9
print.rpart, 10

random.effects, 11
ranef.REEMtree, 10
REEMtree, 7, 11, 14
REEMtree-package, 2
REEMtree.object, 5, 6, 10–12, 13, 15, 16
residuals, 15
residuals.REEMtree, 14
rpart, 12, 14
rpart.object, 16

simpleREEMdata, 15

tree, 16