Package 'rplam'

November 6, 2026

Type Package

Title Robust Estimators for Partially Linear Additive Models

Version 1.0.2

Description Robust estimation for partially linear additive models based on B-splines. The package implements a robust proposal that combines robust linear MM-regression estimators, which use an initial S-estimator followed by an M-estimation step with fixed scale, and B-spline approximations. The M-estimation step uses Tukey's bisquare function. The least-squares counterpart is also implemented. See Boente and Martínez (2023) <doi:10.1016/j.csda.2022.107611> for more details. The package also provides robust and least-squares estimators for additive models, that is, when the covariates enter only the nonparametric component of the model.

License GPL-3

Imports fda, stats, robustbase

Suggests knitr, rmarkdown

Encoding UTF-8

Config/roxygen2/version 8.1.0

NeedsCompilation yes

Author Alejandra Mercedes Martínez [aut, cre]

Maintainer Alejandra Mercedes Martínez <ale_m_martinez@hotmail.com>

Repository CRAN

Date/Publication 2026-11-06 12:30:02 UTC

Contents

am.cl 2
am.rob 4
deviance.am 5
deviance.plam 6
fitted.values.am 6
fitted.values.plam 7
formula.plam 7

my.norm.2 8
plam.cl 9
plam.rob 10
plot.am 12
plot.plam 13
predict.am 14
predict.plam 15
print.plam 15
psi.tukey 16
psi.tukey.derivative 17
psi.w 17
R2 18
R2.rob 19
residuals.am 19
residuals.plam 20
select.nknots.cl 20
select.nknots.cl.am 22
select.nknots.rob 23
select.nknots.rob.am 24
summary.am 25
summary.plam 26
tukey.loss 27

Index **28**

`am.cl` *Classical estimator for additive models*

Description

This function computes the least-squares based estimator for additive models based on B-splines.

Usage

```
am.cl(y, X, np.point = NULL, nknots = NULL, degree.spline = 3)
```

Arguments

`y`	a vector of real numbers.
`X`	a matrix of numbers corresponding to the covariates entering in the additive component of the model.
`np.point`	a matrix for computing the prediction values for the nonparametric part. Must have the same number of columns as X. Defaults to `'NULL'`.
`nknots`	number of internal knots used in the estimation procedure. Defaults to `'NULL'` implies using the BIC criterion of function `select.nknots.cl`.
`degree.spline`	spline degree. Defaults to `'3'`.

Value

A list with the following components:

`fitted.values`	The fitted values.
`g.matrix`	Fitted values of the additive part of the model. The result is a matrix of n x p, with p the number of additive components.
`coeff.const`	Estimation of the intercept.
`coeff.spl`	Estimated coefficients of the B-spline basis.
`nknots`	Number of internal knots selected by the procedure.
`Xpline`	Matrix containing the B-spline basis for each additive function.
`nMat`	Dimension of the B-spline approximation.
`nbasis`	Total number of elements of the basis of B-splines. This corresponds to `d*kj` where `d` is the number of covariates entering in the additive part.
`kj`	Number of elements of the B-spline basis used to approximate each additive function. It is calculated as `nknots + degree.spline`.
`np.prediction`	Vector containing the predicted values obtained for `np.point`.
`y`	Vector of responses.
`X`	Matrix of covariates that enter in the additive part of the model.

Author(s)

Alejandra Mercedes Martínez, <ale_m_martinez@hotmail.com>

References

Boente G. and Martínez A. (2023). A robust spline approach in partially linear additive models. Computational Statistics and Data Analysis, 178, 107611.

Examples

```
set.seed(11)
n <- 100
x1 <- runif(n,-1,1)
x2 <- runif(n,-1,1)
err <- rnorm(n, 0, 0.1)
regre <- 2+x1^3+2*sin(pi*x2)
y <- regre + err
X <- cbind(x1,x2)
sal <- am.cl(y, X)
```

`am.rob` *Robust estimator for additive models*

Description

This function computes a robust estimator for additive models based on B-splines.

Usage

```
am.rob(y, X, np.point = NULL, nknots = NULL, degree.spline = 3, maxit = 100)
```

Arguments

`y`	a vector of real numbers.
`X`	a matrix of numbers corresponding to the covariates entering in the additive component of the model.
`np.point`	a matrix for computing the prediction values for the nonparametric part. Must have the same number of columns as X. Defaults to `'NULL'`.
`nknots`	number of internal knots used in the estimation procedure. Defaults to `'NULL'` implies using the BIC criterion of function `select.nknots.rob`.
`degree.spline`	spline degree. Defaults to `'3'`.
`maxit`	maximum number of iterations for computing the S- and the MM-regression estimators.

Value

A list with the following components:

`fitted.values`	The fitted values.
`g.matrix`	Fitted values of the additive part of the model. The result is a matrix of n x p, with p the number of additive components.
`coeff.const`	Estimation of the intercept.
`coeff.spl`	Estimated coefficients of the B-spline basis.
`nknots`	Number of internal knots selected by the procedure.
`Xpline`	Matrix containing the B-spline basis for each additive function.
`nMat`	Dimension of the B-spline approximation.
`nbasis`	Total number of elements of the basis of B-splines. This corresponds to `d* kj` where `d` is the number of covariates entering in the additive part.
`kj`	Number of elements of the B-spline basis used to approximate each additive function. It is calculated as `nknots + degree.spline`.
`np.prediction`	Vector containing the predicted values obtained for `np.point`.
`y`	Vector of responses.
`X`	Matrix of covariates that enter in the additive part of the model.

Author(s)

Alejandra Mercedes Martínez, <ale_m_martinez@hotmail.com>

References

Boente G. and Martínez A. (2023). A robust spline approach in partially linear additive models. Computational Statistics and Data Analysis, 178, 107611.

Examples

```
set.seed(11)
n <- 100
x1 <- runif(n,-1,1)
x2 <- runif(n,-1,1)
err <- rnorm(n, 0, 0.1)
regre <- 2+x1^3+2*sin(pi*x2)
y <- regre + err
X <- cbind(x1,x2)
sal <- am.rob(y, X)
```

---

deviance.am *Deviance for objects of class* am

Description

This function returns the deviance of the fitted additive model as computed with am.cl or am.rob.

Usage

```
## S3 method for class 'am'
deviance(object, ...)
```

Arguments

object	an object of class am, a result of a call to am.cl or am.rob.
...	additional other arguments. Currently ignored.

Value

A real number.

Author(s)

Alejandra Mercedes Martínez <ale_m_martinez@hotmail.com>

deviance.plam *Deviance for objects of class* plam

Description

This function returns the deviance of the fitted partially linear additive model as computed with plam.cl or plam.rob.

Usage

```
## S3 method for class 'plam'
deviance(object, ...)
```

Arguments

object	an object of class plam, a result of a call to plam.cl or plam.rob.
...	additional other arguments. Currently ignored.

Value

A real number.

Author(s)

Alejandra Mercedes Martínez <ale_m_martinez@hotmail.com>

fitted.values.am *Fitted values for objects of class* am

Description

This function returns the fitted values given the covariates of the original sample under an additive model using a classical or robust estimator computed with am.cl or am.rob.

Usage

```
fitted.values.am(object, ...)
```

Arguments

object	an object of class am, a result of a call to am.cl or am.rob.
...	additional other arguments. Currently ignored.

Value

A vector of fitted values.

Author(s)

Alejandra Mercedes Martínez <ale_m_martinez@hotmail.com>

`fitted.values.plam` *Fitted values for objects of class* `plam`

Description

This function returns the fitted values given the covariates of the original sample under a partially linear additive model using a classical or robust estimator computed with `plam.cl` or `plam.rob`.

Usage

```
fitted.values.plam(object, ...)
```

Arguments

`object`	an object of class `plam`, a result of a call to `plam.cl` or `plam.rob`.
`...`	additional other arguments. Currently ignored.

Value

A vector of fitted values.

Author(s)

Alejandra Mercedes Martínez <ale_m_martinez@hotmail.com>

`formula.plam` *Partially linear additive model formula*

Description

Description of the partially linear additive model formula extracted from an object of class `plam`.

Usage

```
## S3 method for class 'plam'
formula(x, ...)
```

Arguments

`x`	an object of class `plam`, a result of a call to `plam.cl` or `plam.rob`.
`...`	additional other arguments. Currently ignored.

Value

A model formula.

Author(s)

Alejandra Mercedes Martínez <ale_m_martinez@hotmail.com>

`my.norm.2` *Euclidean norm of a vector*

Description

This function calculates the Euclidean norm of a vector.

Usage

```
my.norm.2(x)
```

Arguments

`x`	A real vector.

Value

The Euclidean norm of the input vector.

Author(s)

Alejandra Mercedes Martínez, <ale_m_martinez@hotmail.com>

Examples

```
x <- seq(-2, 2, length=10)
my.norm.2(x)
```

plam.cl *Classical estimator for partially linear additive models*

Description

This function computes the least-squares based estimator for partially linear additive models.

Usage

```
plam.cl(formula, np.point = NULL, nknots = NULL, degree.spline = 3)
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted. For instance, if y is the response vector, Z is a matrix of numbers corresponding to the covariates entering the linear component of the model, and X is a matrix of numbers corresponding to the covariates entering the additive component of the model, the formula is y ~ Z + a(X), where a() is used only as a notation to identify the covariates entering the additive component and does not need to be defined as a function.
np.point	a matrix for computing the prediction values for the nonparametric part. Must have the same number of columns as X. Defaults to 'NULL'.
nknots	number of internal knots used in the estimation procedure. Defaults to 'NULL' implies using the BIC criterion of function select.nknots.cl.
degree.spline	spline degree. Defaults to '3'.

Value

A list with the following components:

fitted.values	The fitted values.
g.matrix	Fitted values of the additive part of the model. The result is a matrix of n x p, with p the number of additive components.
coeff.lin	Estimated coefficients of the linear component (no intercept included)
coeff.const	Estimation of the intercept.
coeff.spl	Estimated coefficients of the B-spline basis.
nknots	Number of internal knots selected by the procedure.
Xpline	Matrix containing the B-spline basis for each additive function.
nMat	Dimension of the B-spline approximation.
nbasis	Total number of elements of the basis of B-splines. This corresponds to d* kj where d is the number of covariates entering in the additive part.
kj	Number of elements of the B-spline basis used to approximate each additive function. It is calculated as nknots + degree.spline.

np.prediction	Vector containing the predicted values obtained for np.point.
y	Vector of responses.
Z	Matrix of 'clean' covariates that enter to the linear part. Categorical variables are converted into dummies.
X	Matrix of covariates that enter in the additive part of the model.

Author(s)

Alejandra Mercedes Martínez, <ale_m_martinez@hotmail.com>

References

Boente G. and Martínez A. (2023). A robust spline approach in partially linear additive models. Computational Statistics and Data Analysis, 178, 107611.

Examples

```
set.seed(11)
n <- 100
z1 <- rnorm(n)
z2 <- rbinom(n, 4, 1/2)
x1 <- runif(n,-1,1)
x2 <- runif(n,-1,1)
err <- rnorm(n, 0, 0.1)
regre <- 2+3*z1-4*z2+x1^3+2*sin(pi*x2)
y <- regre + err
Z <- cbind(z1,z2)
X <- cbind(x1,x2)
sal <- plam.cl(y ~ Z + a(X))
```

---

plam.rob *Robust estimator for partially linear additive models*

Description

This function computes a robust estimator for partially linear additive models using the Tukey's loss function.

Usage

```
plam.rob(
  formula,
  np.point = NULL,
  nknots = NULL,
  degree.spline = 3,
  maxit = 100
)
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted. For instance, if y is the response vector, Z is a matrix of numbers corresponding to the covariates entering the linear component of the model, and X is a matrix of numbers corresponding to the covariates entering the additive component of the model, the formula is y ~ Z + a(X), where a() is used only as a notation to identify the covariates entering the additive component and does not need to be defined as a function.
np.point	a matrix for computing the prediction values for the nonparametric part. Must have the same number of columns as X. Defaults to 'NULL'.
nknots	number of internal knots used in the estimation procedure. Defaults to 'NULL' implies using the BIC criterion of function select.nknots.cl.
degree.spline	spline degree. Defaults to '3'.
maxit	maximum number of iterations for computing the S- and the MM-regression estimators.

Value

A list with the following components:

fitted.values	The fitted values.
g.matrix	Fitted values of the additive part of the model. The result is a matrix of n x p, with p the number of additive components.
sigma.hat	Estimated scale.
coeff.lin	Estimated coefficients of the linear component (no intercept included)
coeff.const	Estimation of the intercept.
coeff.spl	Estimated coefficients of the B-spline basis.
nknots	Number of internal knots selected by the procedure.
Xpline	Matrix containing the B-spline basis for each additive function.
nMat	Dimension of the B-spline approximation.
nbasis	Total number of elements of the basis of B-splines. This corresponds to d* kj where d is the number of covariates entering in the additive part.
kj	Number of elements of the B-spline basis used to approximate each additive function. It is calculated as nknots + degree.spline.
np.prediction	Vector containing the predicted values obtained for np.point.
y	Vector of responses.
Z	Matrix of 'clean' covariates that enter to the linear part. Categorical variables are converted into dummies.
X	Matrix of covariates that enter in the additive part of the model.

Author(s)

Alejandra Mercedes Martínez, <ale_m_martinez@hotmail.com>

References

Boente G. and Martínez A. (2023). A robust spline approach in partially linear additive models. Computational Statistics and Data Analysis, 178, 107611.

Examples

```
set.seed(11)
n <- 100
z1 <- rnorm(n)
z2 <- rbinom(n, 4, 1/2)
x1 <- runif(n,-1,1)
x2 <- runif(n,-1,1)
err <- rnorm(n, 0, 0.1)
regre <- 2+3*z1-4*z2+x1^3+2*sin(pi*x2)
y <- regre + err
Z <- cbind(z1,z2)
X <- cbind(x1,x2)
sal <- plam.rob(y ~ Z + a(X))
```

---

plot.am *Diagnostic plots for objects of class* am

Description

Plot method for class am.

Usage

```
## S3 method for class 'am'
plot(x, which = 1:p, ask = FALSE, ...)
```

Arguments

x	an object of class am, a result of a call to am.cl or am.rob.
which	vector of indices of explanatory variables for which partial residuals plots will be generated. Defaults to all available explanatory variables.
ask	logical value. If TRUE, the graphical device will prompt before going to the next page/screen of output.
...	additional other arguments.

Value

No return value. This function is called for its side effect of producing partial residual plots for the selected explanatory variables.

Author(s)

Alejandra Mercedes Martínez <ale_m_martinez@hotmail.com>

Examples

```
set.seed(11)
n <- 100
x1 <- runif(n,-1,1)
x2 <- runif(n,-1,1)
err <- rnorm(n, 0, 0.1)
regre <- 2+x1^3+2*sin(pi*x2)
y <- regre + err
X <- cbind(x1,x2)
fit.rob <- am.rob(y, X)
plot(fit.rob)
```

---

plot.plam *Diagnostic plots for objects of class* plam

Description

Plot method for class plam.

Usage

```
## S3 method for class 'plam'
plot(x, which = 1:(q + p), ask = FALSE, ...)
```

Arguments

x	an object of class plam, a result of a call to plam.cl or plam.rob.
which	vector of indices of explanatory variables for which partial residuals plots will be generated. Defaults to all available explanatory variables.
ask	logical value. If TRUE, the graphical device will prompt before going to the next page/screen of output.
...	additional other arguments.

Value

No return value. This function is called for its side effect of producing partial residual plots for the selected explanatory variables.

Author(s)

Alejandra Mercedes Martínez <ale_m_martinez@hotmail.com>

Examples

```
set.seed(11)
n <- 100
z1 <- rnorm(n)
z2 <- rbinom(n, 4, 1/2)
x1 <- runif(n,-1,1)
x2 <- runif(n,-1,1)
err <- rnorm(n, 0, 0.1)
regre <- 2+3*z1-4*z2+x1^3+2*sin(pi*x2)
y <- regre + err
Z <- cbind(z1,z2)
X <- cbind(x1,x2)
fit.rob <- plam.rob(y ~ Z + a(X))
plot(fit.rob, which=3)
```

`predict.am` *Fitted values for objects of class* `am`

Description

This function returns the fitted values given the covariates of the original sample under an additive model using a classical or robust estimator computed with `am.cl` or `am.rob`.

Usage

```
## S3 method for class 'am'
predict(object, ...)
```

Arguments

`object`	an object of class `am`, a result of a call to `am.cl` or `am.rob`.
`...`	additional other arguments. Currently ignored.

Value

A vector of fitted values.

Author(s)

Alejandra Mercedes Martínez <ale_m_martinez@hotmail.com>

predict.plam *Fitted values for objects of class* plam

Description

This function returns the fitted values given the covariates of the original sample under a partially linear additive model using a classical or robust estimator computed with plam.cl or plam.rob.

Usage

```
## S3 method for class 'plam'
predict(object, ...)
```

Arguments

object	an object of class plam, a result of a call to plam.cl or plam.rob.
...	additional other arguments. Currently ignored.

Value

A vector of fitted values.

Author(s)

Alejandra Mercedes Martínez <ale_m_martinez@hotmail.com>

print.plam *Print a Partially Linear Additive Model*

Description

The default print method for a plam object.

Usage

```
## S3 method for class 'plam'
print(x, ...)
```

Arguments

x	an object of class plam, a result of a call to plam.cl or plam.rob.
...	additional other arguments. Currently ignored.

Value

The function prints the model formula and returns x invisibly.

Author(s)

Alejandra Mercedes Martínez <ale_m_martinez@hotmail.com>

`psi.tukey` *Derivative of Tukey's bi-square loss function.*

Description

This function evaluates the first derivative of Tukey's bi-square loss function.

Usage

```
psi.tukey(r, k = 4.685)
```

Arguments

`r`	a vector of real numbers
`k`	a positive tuning constant.

Details

This function evaluates the first derivative of Tukey's bi-square loss function.

Value

A vector of the same length as `x`.

Author(s)

Matias Salibian-Barrera, <matias@stat.ubc.ca>, Alejandra Martínez

Examples

```
x <- seq(-2, 2, length=10)
psi.tukey(r=x, k = 1.5)
```

psi.tukey.derivative *Derivative of the Tukey's loss function*

Description

This function calculates the derivative of the Tukey's loss function.

Usage

```
psi.tukey.derivative(x, k = 4.685)
```

Arguments

x	A real vector.
k	tuning constant.

Value

A vector of the same length as x which corresponds to the derivatives of the Tukey's loss function.

Author(s)

Alejandra Mercedes Martínez, <ale_m_martinez@hotmail.com>

Examples

```
x <- seq(-2, 2, length=10)
psi.tukey.derivative(x)
```

psi.w *Tukey's weight function*

Description

This function calculates the Tukey's weight function of the form "Psi(r)/r".

Usage

```
psi.w(r, k = 4.685)
```

Arguments

r	A real vector.
k	tuning constant.

Value

A vector of the same length as r which corresponds to the Tukey's weight function.

Author(s)

Alejandra Mercedes Martínez, <ale_m_martinez@hotmail.com>

Examples

```
r <- seq(-2, 2, length=10)
psi.w(r)
```

R2 *Multiple R-squared*

Description

This function returns the multiple R-squared.

Usage

```
R2(object, ...)
```

Arguments

object	an object of class am, a result of a call to am.cl or am.rob, or class plam, a result of a call to plam.cl or plam.rob
...	additional other arguments. Currently ignored.

Value

A multiple R-squared value.

Author(s)

Alejandra Mercedes Martínez <ale_m_martinez@hotmail.com>

R2.rob *Robust multiple R-squared*

Description

This function returns the multiple R-squared.

Usage

```
R2.rob(object, ...)
```

Arguments

object	an object of class am.rob or plam.rob
...	additional other arguments. Currently ignored.

Value

A robust multiple R-squared value.

Author(s)

Alejandra Mercedes Martínez <ale_m_martinez@hotmail.com>

residuals.am *Residuals for objects of class* am

Description

This function returns the residuals of the fitted additive model using the classical or robust estimators, as computed with am.cl or am.rob.

Usage

```
## S3 method for class 'am'
residuals(object, ...)
```

Arguments

object	an object of class am, a result of a call to am.cl or am.rob.
...	additional other arguments. Currently ignored.

Value

A vector of residuals.

Author(s)

Alejandra Mercedes Martínez <ale_m_martinez@hotmail.com>

`residuals.plam` *Residuals for objects of class* `plam`

Description

This function returns the residuals of the fitted partly linear additive model using the classical or robust estimators, as computed with `plam.cl` or `plam.rob`.

Usage

```
## S3 method for class 'plam'
residuals(object, ...)
```

Arguments

`object`	an object of class `plam`, a result of a call to `plam.cl` or `plam.rob`.
`...`	additional other arguments. Currently ignored.

Value

A vector of residuals.

Author(s)

Alejandra Mercedes Martínez <ale_m_martinez@hotmail.com>

`select.nknots.cl` *Selection of the number of internal knots for the least-squares estimator under a PLAM*

Description

This function automatically selects the number of internal knots for the B-spline approximation. It uses the BIC criterion with the quadratic function.

Usage

```
select.nknots.cl(y, Z, X, degree.spline = 3)
```

Arguments

y	a vector of real numbers.
Z	a matrix of numbers corresponding to the covariates entering linearly into the model.
X	a matrix of numbers corresponding to the covariates entering in the additive component of the model.
degree.spline	spline degree. Defaults to '3'.

Value

A list with the following components:

nknots	Number of internal knots selected by the procedure.
grid.nknots	Grid of internal knots used.
BIC	Values of the BIC criterion for each element of the grid.
kj	Number of elements of the B-spline basis used to approximate each additive function. It is calculated as nknots + degree.spline.
nbasis	Total number of elements of the basis of B-splines. This corresponds to d* kj where d is the number of covariates entering in the additive part.

Author(s)

Alejandra Mercedes Martínez, <ale_m_martinez@hotmail.com>

References

Boente G. and Martínez A. (2023). A robust spline approach in partially linear additive models. Computational Statistics and Data Analysis, 178, 107611.

Examples

```
set.seed(11)
n <- 100
z1 <- rnorm(n)
z2 <- rbinom(n, 4, 1/2)
x1 <- runif(n,-1,1)
x2 <- runif(n,-1,1)
err <- rnorm(n, 0, 0.1)
regre <- 2+3*z1-4*z2+x1^3+2*sin(pi*x2)
y <- regre + err
Z <- cbind(z1,z2)
X <- cbind(x1,x2)
sal <- select.nknots.cl(y, Z, X)
```

`select.nknots.cl.am` *Selection of the number of internal knots for the least-squares estimator for additive models*

Description

This function automatically selects the number of internal knots for the B-spline approximation. It uses the BIC criterion with the quadratic function.

Usage

```
select.nknots.cl.am(y, X, degree.spline = 3)
```

Arguments

`y`	a vector of real numbers.
`X`	a matrix of numbers corresponding to the covariates entering in the additive component of the model.
`degree.spline`	spline degree. Defaults to `'3'`.

Value

A list with the following components:

`nknots`	Number of internal knots selected by the procedure.
`grid.nknots`	Grid of internal knots used.
`BIC`	Values of the BIC criterion for each element of the grid.
`kj`	Number of elements of the B-spline basis used to approximate each additive function. It is calculated as `nknots + degree.spline`.
`nbasis`	Total number of elements of the basis of B-splines. This corresponds to `d* kj` where `d` is the number of covariates entering in the additive part.

Author(s)

Alejandra Mercedes Martínez, <ale_m_martinez@hotmail.com>

References

Boente G. and Martínez A. (2023). A robust spline approach in partially linear additive models. Computational Statistics and Data Analysis, 178, 107611.

Examples

```
set.seed(11)
n <- 100
x1 <- runif(n,-1,1)
x2 <- runif(n,-1,1)
err <- rnorm(n, 0, 0.1)
regre <- 2+x1^3+2*sin(pi*x2)
y <- regre + err
X <- cbind(x1,x2)
sal <- select.nknots.cl.am(y, X)
```

`select.nknots.rob` *Selection of the number of knots for the robust estimator under a PLAM*

Description

This function automatically selects the number of internal knots for the B-spline approximation. It uses the BIC criterion with the Tukey's loss function.

Usage

```
select.nknots.rob(y, Z, X, degree.spline = 3, maxit = 100)
```

Arguments

`y`	a vector of real numbers.
`Z`	a matrix of numbers corresponding to the covariates entering linearly into the model.
`X`	a matrix of numbers corresponding to the covariates entering in the additive component of the model.
`degree.spline`	spline degree. Defaults to `'3'`.
`maxit`	maximum number of iterations for computing the S- and the MM-regression estimators.

Value

A list with the following components:

`nknots`	Number of internal knots selected by the procedure.
`grid.nknots`	Grid of internal knots used.
`RBIC`	Values of the robust BIC criterion for each element of the grid.
`kj`	Number of elements of the B-spline basis used to approximate each additive function. It is calculated as `nknots + degree.spline`.
`nbasis`	Total number of elements of the basis of B-splines. This corresponds to `d* kj` where `d` is the number of covariates entering in the additive part.

Author(s)

Alejandra Mercedes Martínez, <ale_m_martinez@hotmail.com>

References

Boente G. and Martínez A. (2023). A robust spline approach in partially linear additive models. Computational Statistics and Data Analysis, 178, 107611.

Examples

```
set.seed(11)
n <- 100
z1 <- rnorm(n)
z2 <- rbinom(n, 4, 1/2)
x1 <- runif(n,-1,1)
x2 <- runif(n,-1,1)
err <- rnorm(n, 0, 0.1)
regre <- 2+3*z1-4*z2+x1^3+2*sin(pi*x2)
y <- regre + err
Z <- cbind(z1,z2)
X <- cbind(x1,x2)
sal <- select.nknots.rob(y, Z, X)
```

---

select.nknots.rob.am *Selection of the number of knots for the robust estimator for additive models*

Description

This function automatically selects the number of internal knots for the B-spline approximation. It uses the BIC criterion with the Tukey's loss function.

Usage

```
select.nknots.rob.am(y, X, degree.spline = 3, maxit = 100)
```

Arguments

y	a vector of real numbers.
X	a matrix of numbers corresponding to the covariates entering in the additive component of the model.
degree.spline	spline degree. Defaults to '3'.
maxit	maximum number of iterations for computing the S- and the MM-regression estimators.

Value

A list with the following components:

nknots	Number of internal knots selected by the procedure.
grid.nknots	Grid of internal knots used.
RBIC	Values of the robust BIC criterion for each element of the grid.
kj	Number of elements of the B-spline basis used to approximate each additive function. It is calculated as nknots + degree.spline.
nbasis	Total number of elements of the basis of B-splines. This corresponds to d* kj where d is the number of covariates entering in the additive part.

Author(s)

Alejandra Mercedes Martínez, <ale_m_martinez@hotmail.com>

References

Boente G. and Martínez A. (2023). A robust spline approach in partially linear additive models. Computational Statistics and Data Analysis, 178, 107611.

Examples

```
set.seed(11)
n <- 100
x1 <- runif(n,-1,1)
x2 <- runif(n,-1,1)
err <- rnorm(n, 0, 0.1)
regre <- 2+x1^3+2*sin(pi*x2)
y <- regre + err
X <- cbind(x1,x2)
sal <- select.nknots.rob.am(y, X)
```

---

summary.am *Summary for additive models fits*

Description

Summary method for class am.

Usage

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

Arguments

object	an object of class am, a result of a call to am.cl or am.rob.
...	additional other arguments.

Details

This function returns the estimation of the intercept and also the five-number summary and the mean of the residuals for both classical and robust estimators. For the robust estimator it also returns the estimate of the residual standard error.

Value

An object containing the summary of the residuals, as returned by summary. The function also displays the estimated intercept and the coefficient of determination. For robust estimation, it additionally displays the estimated residual standard error.

Author(s)

Alejandra Mercedes Martínez <ale_m_martinez@hotmail.com>

---

summary.plam *Summary for partially linear additive models fits*

Description

Summary method for class plam.

Usage

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

Arguments

object	an object of class plam, a result of a call to plam.cl or plam.rob.
...	additional other arguments.

Details

This function returns the estimation of the intercept, the linear regression coefficients and also the five-number summary and the mean of the residuals for both classical and robust estimators. For the robust estimator it also returns the estimate of the residual standard error.

Value

An object containing the summary of the residuals, as returned by summary. The function also displays the estimated intercept, the linear regression coefficients, and the coefficient of determination. For robust estimation, it additionally displays the estimated residual standard error.

Author(s)

Alejandra Mercedes Martínez <ale_m_martinez@hotmail.com>

`tukey.loss` *Tukey's loss function*

Description

This function computes the Tukey's loss function.

Usage

```
tukey.loss(x, k = 4.685)
```

Arguments

x	A real vector.
k	tuning constant.

Value

A vector of the same length as x which corresponds to values obtained by the Tukey's loss function.

Author(s)

Alejandra Mercedes Martínez, <ale_m_martinez@hotmail.com>

Examples

```
x <- seq(-2, 2, length=10)
tukey.loss(x)
```

Index

am.cl, 2, *5*, *6*, *12*, *14*, *19*, *26*
am.rob, 4, *5*, *6*, *12*, *14*, *19*, *26*

deviance.am, 5
deviance.plam, 6

fitted.values.am, 6
fitted.values.plam, 7
formula.plam, 7

my.norm.2, 8

plam.cl, *6*, *7*, 9, *13*, *15*, *20*, *26*
plam.rob, *6*, *7*, 10, *13*, *15*, *20*, *26*
plot.am, 12
plot.plam, 13
predict.am, 14
predict.plam, 15
print.plam, 15
psi.tukey, 16
psi.tukey.derivative, 17
psi.w, 17

R2, 18
R2.rob, 19
residuals.am, 19
residuals.plam, 20

select.nknots.cl, 20
select.nknots.cl.am, 22
select.nknots.rob, 23
select.nknots.rob.am, 24
summary, *26*
summary.am, 25
summary.plam, 26

tukey.loss, 27