Package ‘slm’

July 23, 2025

Type Package

Title Stationary Linear Models

Version 1.2.0

Author Emmanuel Caron, Jérôme Dedecker, Bertrand Michel

Maintainer Emmanuel Caron <emmanuelcaron3@gmail.com>

Description Provides statistical procedures for linear regression in the general context where the errors are assumed to be correlated. Different ways to estimate the asymptotic covariance matrix of the least squares estimators are available. Starting from this estimation of the covariance matrix, the confidence intervals and the usual tests on the parameters are modified. The functions of this package are very similar to those of 'lm': it contains methods such as summary(), plot(), confint() and predict(). The 'slm' package is described in the paper by E. Caron, J. Dedecker and B. Michel (2019), ``Linear regression with stationary errors: the R package slm'', arXiv preprint <doi:10.48550/arXiv.1906.06583>.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Depends R (>= 2.10)

Collate 'slm-main.R' 'slm.R' 'generative.R' 'auxiliary-fun.R' 'slm-method.R' 'data.R'

Imports ltsa, methods, stats, capushe, sandwich, expm

NeedsCompilation no

Repository CRAN

Date/Publication 2020-08-31 12:00:02 UTC

Contents

`slm-package` *slm: A package for stationary linear models*

Description

The `slm` package enables to fit linear models on datasets considering the dependence between the observations. Most of the functions are based on the functions and methods of `lm`, with the same arguments and the same format for the outputs.

`slm` function, in "slm-main.R"

The `slm` function is the main function of this package. Its architecture is the same as the `lm` function but it takes into account the possible correlation between the observations. To estimate the asymptotic covariance matrix of the least squares estimator, several approaches are available: "fitAR" calls the `cov_AR` function, "spectralproj" the `cov_spectralproj` function, "kernel" the `cov_kernel` function, "efromovich" the `cov_efromovich` function and "select" the `cov_select` function. The "hac" method uses the `sandwich` package, and more precisely, the method described by Andrews (1991) and Zeileis (2004).

Methods for `slm`, in "slm-method.R"

The `slm` function has several associated methods, which are the same as for the `lm` function. The available methods are: `summary`, `confint`, `predict`, `plot` and `vcov`.

Others functions, in "auxiliary-fun.R"

The package has some auxiliary functions, in particular some predefined kernels for the kernel method of `slm` function: the trapeze kernel, the triangle kernel and the rectangular kernel. The user can also define his own kernel and put it in the argument `kernel_fonc` in the `slm` function.

Generative functions, in "generative.R"

The `generative_process` function generates some stationary processes. The `generative_model` function generates some designs.

Data

The package contains a dataset "shan". This dataset comes from a study about fine particle pollution in the city of Shanghai. The data are available on the following website https://archive.ics.uci.edu/ml/datasets/PM2.5+Data+of+Five+Chinese+Cities#.

References

D. Andrews (1991). Heteroskedasticity and autocorrelation consistent covariant matrix estimation. *Econometrica, 59(3), 817-858.*

E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm. *arXiv preprint arXiv:1906.06583*. https://arxiv.org/abs/1906.06583.

A. Zeileis (2004). Econometric computing with HC and HAC covariance matrix estimators.

`confint.slm` *Confidence intervals for the Model Parameters*

Description

Computes confidence intervals for the model parameters.

Usage

```
## S3 method for class 'slm'
confint(object, parm = NULL, level = 0.95, ...)
```

Arguments

`object`	a fitted model object of class `slm`.
`parm`	a vector of integer. Specifies the coordinates of the vector of parameters for which a confidence interval will be given. If missing, all parameters are considered.
`level`	a number between 0 and 1; the confidence level required.
`...`	additional argument(s) for methods.

Value

This function returns the confidence intervals for the parameters of the model.

References

E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm. *arXiv preprint arXiv:1906.06583*. https://arxiv.org/abs/1906.06583.

See Also

confint.lm.

Examples

```
data("shan")
reg1 = slm(shan$PM_Xuhui ~ . , data = shan, method_cov_st = "fitAR", model_selec = -1)
confint(reg1, level = 0.8)

data("co2")
y = as.vector(co2)
x = as.vector(time(co2)) - 1958
reg2 = slm(y ~ x + I(x^2) + I(x^3) + sin(2*pi*x) + cos(2*pi*x) + sin(4*pi*x) +
 cos(4*pi*x) + sin(6*pi*x) + cos(6*pi*x) + sin(8*pi*x) + cos(8*pi*x),
 method_cov_st = "fitAR", model_selec = -1, plot = TRUE)
confint(reg2, level = 0.9)
```

---

cov_AR — *Covariance estimation by AR fitting*

Description

Fit an autoregressive model to the process and compute the theoretical autocovariances of the fitted AR process. By default, the order is chosen by using the AIC criterion (`model_selec = -1`).

Usage

```
cov_AR(epsilon, model_selec = -1, plot = FALSE)
```

Arguments

`epsilon`	numeric vector. An univariate process.
`model_selec`	integer or `-1`. The order of the method, that is the order of the AR process to be fitted on the residuals. If `model_selec = -1`, it is chosen automatically by using the AIC criterion.
`plot`	logical. By default, `plot = FALSE`. If `plot = TRUE`, then the ACF and the PACF of the vector `epsilon` are plotted.

Value

The function returns the vector of the theoretical autocovariances of the AR process fitted on the process `epsilon`.

- `model_selec` the order selected.
- `cov_st` the vector of theoretical autocovariances of the fitted AR process.

References

P.J. Brockwell and R.A. Davis (1991). Time Series: Theory and Methods. *Springer Science & Business Media*.

E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm. *arXiv preprint arXiv:1906.06583*. https://arxiv.org/abs/1906.06583.

Examples

```
x = arima.sim(list(ar=c(0.4,0.2)),1000)
cov_AR(x, model_selec = 2, plot = TRUE)
```

---

cov_efromovich *Spectral density estimation: Efromovich method*

Description

This method estimates the spectral density and the autocovariances of the error process via a lag-window estimator based on the rectangular kernel (see P.J. Brockwell and R.A. Davis (1991). Time Series: Theory and Methods. *Springer Science & Business Media*, page 330). The lag is computed according to Efromovich's algorithm (Efromovich (1998)).

Usage

```
cov_efromovich(epsilon, plot = FALSE)
```

Arguments

- `epsilon` numeric vector. An univariate process.
- `plot` logical. By default, `plot = FALSE`. If `plot = TRUE`, the ACF of the process `epsilon` is plotted.

Value

The function returns the estimated autocovariances of the process, that is the Fourier coefficients of the spectral density estimates, and the order chosen by the algorithm.

- `model_selec` the number of selected autocovariance terms.
- `cov_st` the estimated autocovariances.

References

P.J. Brockwell and R.A. Davis (1991). Time Series: Theory and Methods. *Springer Science & Business Media*.

E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm. *arXiv preprint arXiv:1906.06583*. https://arxiv.org/abs/1906.06583.

S. Efromovich (1998). Data-driven efficient estimation of the spectral density. *Journal of the American Statistical Association*, 93(442), 762-769.

Examples

```
x = arima.sim(list(ar=c(0.4,0.2)),1000)
cov_efromovich(x)
```

---

cov_kernel *Kernel estimation: bootstrap method*

Description

This method estimates the spectral density and the autocovariances of the error process via a lag-window (or kernel) estimator (see P.J. Brockwell and R.A. Davis (1991). Time Series: Theory and Methods. *Springer Science & Business Media*, page 330). The weights are computed according to a kernel K and a bandwidth h (or a lag), to be chosen by the user. The lag can be computed automatically by using a bootstrap technique (as in Wu and Pourahmadi (2009)), via the Rboot function.

Usage

```
cov_kernel(epsilon, model_selec = -1,
 model_max = min(50,length(epsilon)/2), kernel_fonc = triangle,
 block_size = length(epsilon)/2, block_n = 100, plot = FALSE)
```

Arguments

epsilon	numeric vector. An univariate process.
model_selec	integer or -1. The order of the method. If model_selec = -1, the method chooses the treshold automatically. If model_selec = k, then only k autocovariance terms are kept and smoothed by the kernel.
model_max	integer. The maximal order.
kernel_fonc	function. Defines the kernel to use in the method. The user can give his own kernel function.
block_size	integer. If model_selec = -1, it specifies the size of the bootstrap blocks. block_size must be greater than model_max.
block_n	integer. If model_selec = -1, blocks number to use for the bootstrap.
plot	logical. By default, plot = FALSE. If plot = TRUE, the risk curve is returned and the ACF of the process.

Value

The method returns the tapered autocovariance vector with `model_selec` autocovariance terms.

`model_selec`	the number of selected autocovariance terms.
`cov_st`	the estimated autocovariances.

References

E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm. *arXiv preprint arXiv:1906.06583*. https://arxiv.org/abs/1906.06583.

W.B. Wu, M. Pourahmadi (2009). Banding sample autocovariance matrices of stationary processes. *Statistica Sinica*, pp. 1755–1768.

Examples

```
x = arima.sim(list(ar=c(0.7)),1000)
cov_kernel(x, model_selec = -1, block_n = 10, plot = TRUE)
```

`cov_matrix_estimator` *Covariance matrix estimator for slm object*

Description

This function gives the estimation of the asymptotic covariance matrix of the normalized least squares estimator in the case of the linear regression model with strictly stationary errors.

Usage

```
cov_matrix_estimator(object)
```

Arguments

`object`	an object of class `slm`.

Details

The function computes the covariance matrix estimator of the normalized least squares estimator from the vector `cov_st` of a `slm` object. If the user has given the argument `Cov_ST` in the `slm` object, then it is used to compute the final covariance matrix. If the method used is the "hac" method, then the final covariance matrix is computed via the `kernHAC` function of the `sandwich` package, by using the Quadratic Spectral kernel and the bandwidth described in Andrews (1991). For the methods "efromovich", "kernel" and "select", the covariance matrix estimator may not be positive definite. Then we apply the "Positive definite projection" algorithm, which consists in replacing all eigenvalues lower or equal to zero with the smallest positive eigenvalue of the covariance matrix.

Value

This function returns the estimation of the asymptotic covariance matrix of the normalized least squares estimator.

References

D. Andrews (1991). Heteroskedasticity and autocorrelation consistent covariant matrix estimation. *Econometrica, 59(3), 817-858.*

E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm. *arXiv preprint arXiv:1906.06583*. https://arxiv.org/abs/1906.06583.

A. Zeileis (2004). Econometric computing with HC and HAC covariance matrix estimators.

See Also

The R package `sandwich`.

`kernHAC` for HAC methods.

`cov_method` *Methods to estimate the autocovariances of a process*

Description

This function gives the estimation of the autocovariances of the error process, with the method chosen by the user. Five methods are available: "fitAR", "spectralproj", "efromovich", "kernel" and "select".

Usage

```
cov_method(epsilon, method_cov_st = "fitAR", model_selec = -1,
  model_max = NULL, kernel_fonc = NULL, block_size = NULL,
  block_n = NULL, plot = FALSE)
```

Arguments

`epsilon`	numeric vector. An univariate process.
`method_cov_st`	the method chosen by the user to estimate the autocovariances of the error process. The user has the choice between the methods "fitAR", "spectralproj", "efromovich", "kernel", "select" or "hac". By default, the "fitAR" method is used.
`model_selec`	integer or `-1`. The order of the method. If `model_selec = -1`, the method works automatically.
`model_max`	integer. Maximal dimension of the method.
`kernel_fonc`	function. Use this argument if `method_cov_st = kernel`. Define the kernel to use in the method. The user can give his own kernel function.

block_size	integer. Size of the bootstrap blocks if method_cov_st = kernel. block_size must be greater than model_max.
block_n	integer. Blocks number to use for the bootstrap if method_cov_st = kernel.
plot	logical. By default, plot = FALSE.

Value

The function returns the autocovariances computed with the chosen method.

References

E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm. *arXiv preprint arXiv:1906.06583*. https://arxiv.org/abs/1906.06583.

Examples

```
x = arima.sim(list(ar=c(0.4,0.2)),1000)
cov_method(x, method_cov_st = "fitAR", model_selec = -1)
```

---

cov_select *Covariances Selection*

Description

Allows the user to select the lags of the autocovariance terms of the process to be kept.

Usage

```
cov_select(epsilon, model_selec, plot = FALSE)
```

Arguments

epsilon	numeric vector. An univariate process.
model_selec	a vector with the positive lags of the selected autocovariance terms. The variance (lag = 0) is automatically selected.
plot	logical. By default, plot = FALSE. If plot = TRUE the ACF of the process is plotted.

Details

In the framework of slm, this is a manual method for estimating the covariance matrix of the error process by only selecting some autocovariance terms from the residual autocovariances.

Value

This function returns the estimated autocovariance terms.

model_selec	the vector with the positive lag of the selected autocovariance terms.
cov_st	the vector of the selected autocovariances.

References

E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm. *arXiv preprint arXiv:1906.06583*. https://arxiv.org/abs/1906.06583.

Examples

```
x = arima.sim(list(ar=c(0.2,0.1,0.25)),1000)
cov_select(x, c(1,3,5))
```

---

cov_spectralproj *Data-driven spectral density estimation*

Description

Computes a data-driven histogram estimator of the spectral density of a process and compute its Fourier coefficients, that is the associated autocovariances. For a dimension d, the estimator of the spectral density is an histogram on a regular basis of size d. Then we use a penalized criterion in order to choose the dimension which balance the bias and the variance, as proposed in Comte (2001). The penalty is of the form $c * d/n$, where c is the constant and n the sample size. The dimension and the constant of the penalty are chosen with the slope heuristic method, with the dimension jump algorithm (from package "capushe").

Usage

```
cov_spectralproj(epsilon, model_selec = -1,
 model_max = min(100,length(epsilon)/2), plot = FALSE)
```

Arguments

`epsilon`	numeric vector. An univariate process.
`model_selec`	integer. The dimension of the method. If `model_selec = -1`, the method works automatically and take a dimension between 1 and `model_max`.
`model_max`	integer. The maximal dimension. By default, it is equal to the minimum between 100 and the length of the process divided by 2.
`plot`	logical. By default, `plot = FALSE`. If `plot = TRUE`, plot the spectral density estimator of the process.

Value

The function returns the estimated autocovariances of the process, that is the Fourier coefficients of the spectral density estimate, and the dimension chosen by the algorithm.

`model_selec`	the dimension selected.
`cov_st`	the estimated autocovariances.

References

J.P. Baudry, C. Maugis B. and Michel (2012). Slope heuristics: overview and implementation. *Statistics and Computing*, 22(2), 455–470.

E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm. *arXiv preprint arXiv:1906.06583*. https://arxiv.org/abs/1906.06583.

F. Comte (2001). Adaptive estimation of the spectrum of a stationary Gaussian sequence. *Bernoulli*, 7(2), 267-298.

See Also

The R package capushe.

Slope heuristic algorithm DDSE.

Dimension jump algorithm Djump.

Examples

```
x = arima.sim(list(ar=c(0.2), ma=c(0.3,0.05)), n=100)
cov_spectralproj(x, model_selec = -1)
```

---

generative_model *Some linear model*

Description

This function returns a design for the regression linear model, without the intercept. The user can choose one of the two models: "mod1" or "mod2". The first model "mod1" contains just one column, equal to $i^2 + X_i$, $i = 1, ..., n$, where X is an AR(1) process with `phi_1 = 0.5`.

The second model "mod2" contains two columns, the first equal to $log(i) + sin(i) + X_i$ and the second equal to i, for $i = 1, ..., n$. The process X is again an AR(1) process with `phi_1 = 0.5`. More information about "mod2" is available in the paper of E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm.

Usage

```
generative_model(n, model = "mod1")
```

Arguments

`n`	integer. The sample size.
`model`	a list of character "mod1" or "mod2" to choose the model.

Value

This function returns a data-frame which contains a simulated random design.

References

E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm. *arXiv preprint arXiv:1906.06583*. https://arxiv.org/abs/1906.06583.

Examples

```
generative_model(500,"mod1")
```

---

generative_process *Some stationary processes*

Description

This is a generative function. The user chooses one of the `process`: "iid", "AR1", "AR12", "MA12", "Nonmixing", "sysdyn", and it generates the chosen process. These processes are fully described in the paper of E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm. *arXiv preprint arXiv:1906.06583*. https://arxiv.org/abs/1906.06583.

Usage

```
generative_process(n, process = "AR1", phi = "numeric",
  theta = "numeric")
```

Arguments

`n`	integer. The sample size.
`process`	a list of character to choose the process.
`phi`	a numeric vector with AR parameters if the process is "AR1" or "AR12".
`theta`	a numeric vector with MA parameters if the process is "MA12".

Value

This function returns a vector of observations drawn according to the selected process.

References

E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm. *arXiv preprint arXiv:1906.06583*. https://arxiv.org/abs/1906.06583.

Examples

```
generative_process(200,"Nonmixing")
```

plot.slm *Plot.slm*

Description

Same function as the plot.lm function.

Usage

```
## S3 method for class 'slm'
plot(x, ...)
```

Arguments

x	slm object.
...	other parameters to be passed through to plotting functions.

Value

This function returns the graphics of plot.lm(x).

Examples

```
data("shan")
reg = slm(shan$PM_Xuhui ~ . , data = shan, method_cov_st = "fitAR", model_selec = -1)
plot(reg)
```

predict.slm *Predict for slm object*

Description

Predicted values based on slm object.

Usage

```
## S3 method for class 'slm'
predict(object, newdata = NULL, interval = "confidence",
  level = 0.95, ...)
```

Arguments

- `object` — an object of class `slm`.
- `newdata` — an optional data frame in which to look for variables with which to predict. `newdata` must contain only variables and not the intercept. If omitted, the fitted values are used.
- `interval` — type of interval calculation. It can be only `interval = "confidence"`, the default value. It computes the confidence intervals for $x'beta$, where x' is a new observation of the design.
- `level` — a number between 0 and 1, which indicates the tolerance/confidence level.
- `...` — further arguments passed to or from other methods.

Details

This function produces predicted values, obtained by evaluating the regression function in the frame `newdata` (which defaults to `model.frame(object)`). If `newdata` is omitted the predictions are based on the data used for the fit.

Value

This function produces a vector of predictions or a matrix of predictions and bounds with column names `fit`, `lwr`, and `upr` if `interval` is set.

See Also

`predict.lm`.

Examples

```
data("shan")
reg1 = slm(shan$PM_Xuhui ~ . , data = shan, method_cov_st = "fitAR", model_selec = -1)
predict(reg1)

data("co2")
y = as.vector(co2)
x = as.vector(time(co2)) - 1958
reg2 = slm(y ~ x + I(x^2) + I(x^3) + sin(2*pi*x) + cos(2*pi*x) + sin(4*pi*x) +
 cos(4*pi*x) + sin(6*pi*x) + cos(6*pi*x) + sin(8*pi*x) + cos(8*pi*x),
 method_cov_st = "fitAR", model_selec = -1)
predict(reg2)
```

`Rboot` *Risk estimation for a tapered covariance matrix estimator via bootstrap method*

Description

This function computes an estimation of the risk for the tapered covariance matrix estimator of a process via a bootstrap method, for a specified treshold and a specified kernel.

Usage

```
Rboot(epsilon, treshold, block_size, block_n, model_max, kernel_fonc)
```

Arguments

`epsilon`	numeric vector. An univariate process.
`treshold`	integer. Number of estimated autocovariance terms that we consider for the estimation of the covariance matrix.
`block_size`	integer. The size of the bootstrap blocks. `block_size` must be greater than `model_max`.
`block_n`	integer. Blocks number used for the bootstrap.
`model_max`	integer. The maximal dimension, that is the maximal number of terms available to estimate the covariance matrix.
`kernel_fonc`	function. The kernel to use. The user can define his own kernel and put it in the argument.

Value

This function returns a list with:

`risk`	for one treshold, the value of the estimated risk.
`SE`	the standard-error due to the bootstrap.

References

E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm. *arXiv preprint arXiv:1906.06583*. https://arxiv.org/abs/1906.06583.

rectangle *Rectangular kernel*

Description

Rectangular kernel

Usage

```
rectangle(x)
```

Arguments

x	a vector of real numbers.

Value

This function computes the values of the rectangular kernel at points x.

Examples

```
x = seq(-2,2,length=1000)
y = rectangle(x)
plot(x,y)
```

shan *PM2.5 Data of Shanghai*

Description

This dataset comes from a study about fine particle pollution in five Chinese cities. The data are available on the following website https://archive.ics.uci.edu/ml/datasets/PM2.5+Data+of+Five+Chinese+Cities#. The present dataset concerns the city of Shanghai. From the initial dataset, we have removed the lines that contain NA observations and we then extract the first 5000 observations. Then we consider only pollution variables and weather variables.

Usage

```
data("shan")
```

Format

A data frame with 5000 rows and 10 variables:

PM_Xuhui PM2.5 concentration in the Xuhui district ($ug/m3$).

PM_Jingan PM2.5 concentration in the Jing'an district ($ug/m3$).

PM_US.Post PM2.5 concentration in the U.S diplomatic post ($ug/m3$).

DEWP Dew Point ($CelsiusDegree$).

TEMP Temperature ($CelsiusDegree$).

HUMI Humidity ($\%$).

PRES Pressure (hPa).

Iws Cumulated wind speed (m/s).

precipitation hourly precipitation (mm).

Iprec Cumulated precipitation (mm).

References

E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm. *arXiv preprint arXiv:1906.06583*. https://arxiv.org/abs/1906.06583.

X. Liang, S. Li, S. Zhang, H. Huang, S.X. Chen (2016). PM2.5 data reliability, consistency, and air quality assessment in five Chinese cities. *Journal of Geophysical Research: Atmospheres*, 121(17), 10–220.

`slm` *Fitting Stationary Linear Models*

Description

`slm` is used to fit linear models when the error process is assumed to be strictly stationary.

Usage

```
slm(myformula, data = NULL, model = TRUE, x = FALSE, y = FALSE,
  qr = TRUE, method_cov_st = "fitAR", cov_st = NULL, Cov_ST = NULL,
  model_selec = -1, model_max = 50, kernel_fonc = NULL,
  block_size = NULL, block_n = NULL, plot = FALSE)
```

Arguments

`myformula`	an object of class "`formula`" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`data`	an optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the variables in the model. If not found in `data`, the variables are taken from `environment(formula)`, typically the environment from which `slm` is called.

model, x, y, qr	logicals. If TRUE the corresponding components of the fit (the model frame, the model matrix, the response, the QR decomposition) are returned.
method_cov_st	the method chosen by the user to estimate the autocovariances of the error process. The user has the choice between the methods "fitAR", "spectralproj", "efromovich", "kernel", "select" or "hac". By default, the "fitAR" method is used.
cov_st	numeric vector. The estimated autocovariances of the error process. The user can give his own vector.
Cov_ST	matrix. It is an argument given by the user if he wants to use his own covariance matrix estimator.
model_selec	integer or -1. The order of the method. If model_selec = -1, the method works automatically.
model_max	integer. Maximal order of the method.
kernel_fonc	function. Use this argument if method_cov_st = kernel. Define the kernel to use in the method. The user can give his own kernel function.
block_size	integer. Size of the bootstrap blocks if method_cov_st = kernel. block_size must be greater than model_max.
block_n	integer. Blocks number to use for the bootstrap if method_cov_st = kernel.
plot	logical. By default, plot = FALSE.

Details

The slm function is based on the architecture of the lm function. Models for slm 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. See the documentation of lm for more details.

Value

slm returns an object of class "slm". The function summary is used to obtain and print a summary of the results. The generic accessor functions coefficients, effects, fitted.values and residuals extract various useful features of the value returned by slm. An object of class "slm" is a list containing at least the following components:

method_cov_st	print the method chosen.
cov_st	the estimated autocovariances of the error process. NA if "hac" is used.
Cov_ST	if given by the user, the estimated covariance matrix of the error process. NA if "hac" is used.
model_selec	the order of the method.
norm_matrix	the normalization matrix of the least squares estimator.
design_qr	the matrix $(X^t X)^{-1}$.
coefficients	a named vector of the estimated coefficients.
residuals	the residuals, that is response minus fitted values.
fitted.values	the fitted values.

rank	the numeric rank of the fitted linear model.
df.residual	the number of observations minus the number of variables.
call	the matched call.
terms	the terms object used.
xlevels	(only where relevant) a record of the levels of the factors used in fitting.
y	if requested, the response used.
x	if requested, the model matrix used.
model	if requested (the default), the model frame used.

References

E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm. *arXiv preprint arXiv:1906.06583*. https://arxiv.org/abs/1906.06583.

See Also

summary for summaries.

The generic functions coef, effects, residuals, fitted, vcov.

predict for prediction, including confidence intervals for $x'beta$, where x' is a new observation of the design.

confint for confidence intervals of *parameters*.

Examples

```
data("shan")
slm(shan$PM_Xuhui ~ . , data = shan, method_cov_st = "fitAR", model_selec = -1)

data("co2")
y = as.vector(co2)
x = as.vector(time(co2)) - 1958
reg1 = slm(y ~ x + I(x^2) + I(x^3) + sin(2*pi*x) + cos(2*pi*x) + sin(4*pi*x) +
 cos(4*pi*x) + sin(6*pi*x) + cos(6*pi*x) + sin(8*pi*x) + cos(8*pi*x),
 method_cov_st = "fitAR", model_selec = -1, plot = TRUE)

reg2 = slm(y ~ x + I(x^2) + I(x^3) + sin(2*pi*x) + cos(2*pi*x) + sin(4*pi*x) +
 cos(4*pi*x) + sin(6*pi*x) + cos(6*pi*x) + sin(8*pi*x) + cos(8*pi*x),
 method_cov_st = "kernel", model_selec = -1, model_max = 50, kernel_fonc = triangle,
 block_size = 100, block_n = 100)
```

slm-class *slm class*

Description

An S4 class to create an slm object.

Slots

method_cov_st the method used to compute the autocovariance vector of the error process. The user has the choice between the methods "fitAR", "spectralproj", "efromovich", "kernel", "select" or "hac". By default, the "fitAR" method is used.

cov_st a numeric vector with the estimated autocovariances of the error process, computed from the method_cov_st method.

Cov_ST the estimated covariance matrix of the error process, computed from the method_cov_st method.

model_selec integer. The order of the chosen method. If model_selec = -1, the method works automatically.

norm_matrix the normalization matrix of the design X.

design_qr the matrix $(X^t X)^{-1}$.

References

E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm. *arXiv preprint arXiv:1906.06583*. https://arxiv.org/abs/1906.06583.

summary.slm *Summarizing Stationary Linear Model Fits*

Description

Summary method for class "slm".

Usage

```
## S3 method for class 'slm'
summary(object, correlation = FALSE,
  symbolic.cor = FALSE, ...)
```

Arguments

- `object` — an object of class "slm", usually, a result of a call to `slm`.
- `correlation` — logical. If `TRUE`, the correlation matrix of the estimated parameters is returned and printed.
- `symbolic.cor` — logical. If `TRUE`, print the correlations in a symbolic form (see `symnum`) rather than as numbers.
- `...` — further arguments passed to or from other methods.

Value

The function `summary.slm` computes and returns a list of summary statistics of the fitted linear model given in `object`, using the components (list elements) "call" and "terms" from its argument, plus:

- `residuals` — the residuals, that is response minus fitted values.
- `coefficients` — a $p * 4$ matrix with columns for the estimated coefficient, its standard error, z-statistic and corresponding (two-sided) p-value. Aliased coefficients are omitted.
- `aliased` — named logical vector showing if the original coefficients are aliased.
- `sigma` — the square root of the estimated variance of the error process.
- `df` — degrees of freedom, a 3-vector $(p, n - p, p*)$, the first being the number of non-aliased coefficients, the last being the total number of coefficients.
- `chi2statistic` — a 2-vector with the value of the chi2-statistic with its degree of freedom.
- `r.squared` — R^2, the 'fraction of variance explained by the model'.
- `cov.unscaled` — the matrix $(X^t X)^{-1}$.
- `correlation` — the correlation matrix corresponding to the above `cov.unscaled`, if `correlation = TRUE` is specified.
- `symbolic.cor` — (only if `correlation` is true.) The value of the argument `symbolic.cor`.

References

E. Caron, J. Dedecker and B. Michel (2019). Linear regression with stationary errors: the R package slm. *arXiv preprint arXiv:1906.06583*. https://arxiv.org/abs/1906.06583.

See Also

The model fitting function `slm`, `summary`.

The function `coef` extracts the matrix of coefficients with standard errors, z-statistics and p-values.

Examples

```
data("shan")
reg1 = slm(shan$PM_Xuhui ~ . , data = shan, method_cov_st = "fitAR", model_selec = -1)
summary(reg1)

data("co2")
```

```
y = as.vector(co2)
x = as.vector(time(co2)) - 1958
reg2 = slm(y ~ x + I(x^2) + I(x^3) + sin(2*pi*x) + cos(2*pi*x) + sin(4*pi*x) +
 cos(4*pi*x) + sin(6*pi*x) + cos(6*pi*x) + sin(8*pi*x) + cos(8*pi*x),
 method_cov_st = "fitAR", model_selec = -1)
summary(reg2)
```

`trapeze` *Trapeze kernel*

Description

Trapeze kernel

Usage

```
trapeze(x, width = 0.8)
```

Arguments

`x`	a vector of real numbers.
`width`	a number between 0 and 1.

Value

This function computes the values of the trapeze kernel at points `x`.

Examples

```
x = seq(-2,2,length=1000)
y = trapeze(x, width=0.5)
plot(x,y)
```

`triangle` *Kernel triangle*

Description

Kernel triangle

Usage

```
triangle(x)
```

Arguments

`x`	a vector of real numbers.

Value

This function computes the values of the triangle kernel at points x.

Examples

```
x = seq(-2,2,length=1000)
y = triangle(x)
plot(x,y)
```

---

vcov.slm *Calculate Variance-Covariance Matrix for a Fitted Model Object of class slm*

Description

Returns the variance-covariance matrix of the (non-normalized) least squares estimators for an object of class slm.

Usage

```
## S3 method for class 'slm'
vcov(object, ...)
```

Arguments

object	a fitted model object of class slm.
...	additional arguments for method functions.

Value

The variance-covariance matrix of the (non-normalized) least squares estimators for an object of class slm.

See Also

The generic function vcov.

The function cov_matrix_estimator.

Examples

```
n = 500
eps = generative_process(n,"AR1",c(0.7))
X = as.matrix(generative_model(n,"mod2"))
Y = 3 + 2*X[,2] + eps
reg = slm(Y ~ X, method_cov_st = "fitAR", model_selec = -1)
vcov(reg)
```

Index