Package 'regg'

May 16, 2024

Type Package

Title Extensible Grammar for Regression Analysis

Version 0.0.1

Maintainer Christopher Mann <cmann3@unl.edu>

Description An ecosystem and grammar, based on the tidyverse, for conducting regression analysis and easily developing regression methods, statistics, and tests. The framework is designed to assist users by offering enhanced model selection tools using non-standard evaluation, allowing multiple regression models to be easily estimated and compared, and a simple method for changing standard errors, including statistics, and performing tests.

Depends generics

Imports dplyr,
ggplot2,
rlang,
stats,
tibble

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

R topics documented:

`add_stat` *Add a Statistic to* `regg` *Object*

Description

This function is used to add a statistic function to a `ggr` object. It is intended to be called by the `regg` method of the original regression statistic function.

Usage

```
add_stat(x, stat, ...)
```

Arguments

`x`	object of class `regg` or `rgo_model`
`stat`	function for estimating the statistic, to be applied to the regression model
`...`	objects passed to methods

Value

the object `x`

add_test *Add a Test to* regg *Object*

Description

This function is used to add a test function to a ggr object. It is intended to be called by the regg method of the original regression test function.

Usage

```
add_test(x, test, ...)
```

Arguments

x	object of class regg or rgo_model
test	function for estimating the test, to be applied to the regression model
...	objects passed to methods
args	a quoted list of function arguments to be passed instead of ...

Value

the object x

add_x *Estimate a New Regression by Adding Independent Variables*

Description

Builds a new regression model using the regression in x as the basic template, including the model terms. Unnamed objects passed to add_x are included as additional independent variables, using grammar from reg_select. Variables from the previous regression can be removed with the - prefix.

Usage

```
add_x(x, ...)
```

Arguments

x	rgo object containing a regression model.
...	If unnamed, variables to be added as independent variables in the new regression using reg_select. If named, additional fields passed to regg.

Value

environment of class regg

`as_lambda` *Create purrr-style Functions from R Objects*

Description

Function creation from formulas, calls, and other R objects, similar to `as_mapper`.

Usage

```
as_lambda(x, envir = parent.frame(), ...)
```

Arguments

`x`	A function, formula, call, or vector
`envir`	environment associated with created function
`...`	arguments passed to methods.

Details

`as_lambda` is a convenience function used to quickly create functions for use in functions throughout the `regg` package, similar to lambdas in the `purrr` package. However, there are some minor differences between the two, as described below, due to how lambdas are used in `regg`.

When a function is passed to `as_lambda`, it is returned as is.

When a formula is passed to `as_lambda`, e.g. `~ .x - 5`, the object on the right-hand side of `~` becomes the function body with any variable prefaced with a period `.` following by a letter, e.g. `.a` or `.z`, becoming an argument name. The argument names are sorted so that `~ .b - .a` becomes `function(.a, .b, ...) .b - .a`. Note that `..1`, `...2`, etc. can be used to reference objects beyond those explicitly labelled. Therefore, `as_lambda` provides more flexibility when creating functions.

Two-sided formula can also be used. In this case, the objects on the left-hand side of `~`, separated by a `+`, are interpreted as the variable names in the resulting functions. For example, `i ~ i + 1` is equivalent to `function (i, ...) i + 1`. Similarly, `i + j ~ i * j + 1` is equivalent to `function(i, j, ...) i * j + 1`.

Calls that are passed to `as_lambda` are treated the same as the right-hand side of a formula. `as_lambda( substitute(sqrt(.x)) )` is equivalent to `function(.x, ...) sqrt(.x)`

One of the main uses of lambdas in `regg` is to rename model variables using `:=` inside of `reg_select`. One way that this shows up is how `as_lambda` handles numeric vectors. As with `as_mapper`, numeric vectors are converted to functions that extract elements specified by the numeric vector. The difference is how it handles single, character values. Here, it will extract the individual characters. For example, `as_lambda(2:4)("hello")` will return `"ell"`.

Value

'function' object

Examples

```
as_lambda(\(i) i + 1)
as_lambda(seq(1,5, by = 2))
as_lambda(x + y ~ x[1] * y)
as_lambda(~ .x + .y)
```

as_stat *Create an* rgo_test *Object*

Description

rgo_test objects are lists containing statistics, associated degrees of freedom, p-values, and other fields containing relevant test information.

Usage

```
as_stat(x, label = NULL, ...)
```

Arguments

label	character name to display when printed
...	Other objects included in the attributes.
stat, df, pval	numeric values or vectors

Value

list of class "rgo_test"

as_test *Create an* rgo_test *Object*

Description

rgo_test objects are lists containing statistics, associated degrees of freedom, p-values, and other fields containing relevant test information.

Usage

```
as_test(stat, pval = NULL, df = NULL, label = NULL, ...)
```

Arguments

stat, df, pval	numeric values or vectors
label	character name to display when printed
...	Other objects include in the list.

Value

list of class "rgo_test"

extract *Extract Elements from Regression Models*

Description

Extract Elements from Regression Models

Usage

```
## S3 method for class 'rgo_model'
x$name

## S3 method for class 'rgo_model'
x[[i, inherits = FALSE, ...]]
```

Arguments

x	object of class rgo
name, i	character describing object to extract
inherits	should earlier models be searched? Defaults to TRUE when using `$`, and FALSE when using `[[`.
...	objects passed to methods

Value

R object

extract_reg *Extract Regression Objects*

Description

Obtain common regression objects such as the coefficients, fitted values, and residuals.

Usage

```
## S3 method for class 'rgo_model'
coef(object, write = TRUE, ...)

## S3 method for class 'rgo_model'
fitted(object, write = TRUE, ...)

## S3 method for class 'rgo_model'
residuals(
  object,
  type = c("working", "response", "deviance", "pearson", "partial"),
  write = TRUE,
  ...
)
```

```
## S3 method for class 'rgo_model'
weights(object, write = TRUE, ...)
```

Arguments

write	should the requested object be inserted in x, if it not there
...	objects passed to methods
x	regg or rgo_model object

Value

the object, extracted from x

---

find_else *Find Object in Regression Model Else Evaluate Expression*

Description

These functions look for an object within a rgo object. If not found, an expression, expr, will be evaluated and returned. The result may be written to the object so that it is easily available in the future. fit_find attempts to fit a regression model first - if it is unfitted - then searches for the object. These functions are intended for regg developers.

Usage

```
find_else(x, what, expr, mode = "any", inherits = TRUE, write = TRUE)

fit_find(x, what, expr, mode = "any", inherits = FALSE, write = TRUE, ...)
```

Arguments

x	object of class rgo to be searched
what	character vector to search
expr	expression to be evaluated if object is not found
mode	the mode or type of object sought. See details in exists.
inherits	should previous models be searched?
write	should the evaluated expression be inserted in x as what?
...	objects passed to fit.rgo_model

Value

R object

find_models *Return All Nested Models in* rgo *Regression Object*

Description

Return All Nested Models in rgo Regression Object

Usage

```
find_models(x)
```

Arguments

x	object of class rgo_m to be searched

Value

list containing objects containing class rgo_m

fit *Fit a* regg *Regression Model*

Description

Fit a regg Regression Model

Usage

```
## S3 method for class 'regg'
fit(x, ...)

## S3 method for class 'rgo_model'
fit(x, refit = FALSE, keep = FALSE, ...)

## S3 method for class 'regg'
refit(x, ...)

## S3 method for class 'rgo_model'
refit(x, ...)
```

Arguments

x	object to be fitted
...	objects passed to methods
refit	should the model be fitted again if it is already fitted?
keep	should the X and y matrices be kept after fitting?

Details

regg and rgo_model objects are initialized with a flag "is_fitted" set to FALSE. ggr_fit first checks whether this flag is TRUE. If so, the object is returned, unless the refit argument is set to TRUE. If the flag is FALSE, then the following takes place.

First, get_model_matrix is run to set the X and y variable fields inside of the environment. Next, the method of fitting the model is called on the ggr_model object and the "is_fitted" flag is set to TRUE. Any post-fit functions are applied to model, the standard errors and other regression statistics are calculated, then statistical tests are performed.

Value

ggr_model or ggreg object

---

fit_ols *Fit a Model using Ordinary Least Squares*

Description

This is used by ols and related methods. It is not intended to be used directly for analysis, but within a regression method to compute the relevant QR matrix and coefficients, and add the appropriate fields to the model object. It is a wrapper for lm.fit and lm.wfit.

Usage

```
fit_ols(model, X, y, weights = NULL, offset = NULL, intercept = NULL)
```

Arguments

model	object of class rgo_model
X	numeric matrix of independent variables
y	numeric vector or matrix of predictor variables
weights	numeric vector of weights or NULL
offset	numeric vector to specify a priori known component to be included in the predictor.
intercept	does the model include an intercept? If NULL, fit_ols will attempt to determine automatically.

Value

the regression model

`get_component` *Obtain & Extract Regression Model Components*

Description

These functions are intended to be used inside of methods to extract regression components - such as the X or y matrices - for fitting.

Usage

```
get_cooks(x)

get_df(x, write = TRUE)

get_fitted(x)

get_hat(x)

get_model_matrix(x, write = TRUE)

get_offset(x, write = TRUE)

get_resids(x)

get_weights(x, write = TRUE)

get_x(x, write = TRUE)

get_y(x, write = TRUE)

has_intercept(x, write = TRUE)
```

Arguments

`x`	`rgo_model` object
`write`	should the requested object be inserted in `x`, if it not found
`...`	objects passed to methods

Value

the requested object, usually numeric vector or matrix

Functions

- `get_cooks()`: Obtain the cook's distance.
- `get_df()`: Obtain the number of degrees of freedom of the residual
- `get_fitted()`: Obtain the regression fitted values
- `get_hat()`: Obtain the leverage values for the regression.
- `get_model_matrix()`: Obtain the model matrix associated with `reg_select`.

- `get_offset()`: Obtain the regression offset
- `get_resids()`: Obtain the regression residuals.
- `get_weights()`: Obtain the vector of weights
- `get_x()`: Obtain the X matrix of independent variables
- `get_y()`: Obtain the y matrix or vector of dependent variables
- `has_intercept()`: Does the model contain an intercept?

`get_data` *Find Regression Data*

Description

These functions look for a data set within an `rgo` object. If used on an `rgo_model`, the underlying data set - subset if specified - will be returned.

Usage

```
get_data(x, ...)
```

Arguments

`x`	object of class `rgo` to be searched
`...`	objects passed to methods

Value

data.frame or related object

`get_stats` *Obtain & Extract Regression Statistics*

Description

These functions are intended to be used inside of methods to extract regression statistics - such as the residual sum of squares (rss) - for calculating other statistics or tests.

Usage

```
get_mss(x, write = TRUE)

get_rank(x, write = TRUE)

get_rss(x, write = TRUE)

get_se(x, write = TRUE)
```

Arguments

x	rgo_model object
write	should the requested object be inserted in x, if it not found

Value

the requested object, usually numeric vector or matrix

Functions

- get_mss(): Obtain the mean sum of squares
- get_rank(): Obtain the rank of the regression
- get_rss(): Obtain the sum of squared residuals
- get_se(): Obtain the standard errors for the model coefficients

---

influence_regg *regg Deletion Diagnostics & Influence Measures*

Description

Compute leave-one-out deletion diagnostics for regg models. See influence.measures from the stats package.

Usage

```
## S3 method for class 'rgo_model'
cooks.distance(model, write = TRUE, ...)

## S3 method for class 'rgo_model'
dfbeta(model, write = TRUE, ...)

## S3 method for class 'rgo_model'
dfbetas(model, write = TRUE, ...)

## S3 method for class 'rgo_model'
hatvalues(model, write = TRUE, ...)

## S3 method for class 'rgo_model'
influence(model, write = TRUE, ...)

## S3 method for class 'rgo_model'
rstandard(model, write = TRUE, type = c("sd.1", "predictive"), ...)

## S3 method for class 'rgo_model'
rstudent(model, write = TRUE, ...)

## S3 method for class 'rgo_model'
deviance(object, write = FALSE, ...)
```

Arguments

model	rgo_model object
write	should the requested object be inserted in x, if it not there
...	objects passed to methods

Value

the object, extracted from x

---

is_regg *Check Whether Input is a 'regg' Object*

Description

Check Whether Input is a 'regg' Object

Usage

```
is_regg(x)
```

Arguments

x	R object to be tested

Value

TRUE if x inherits "regg", otherwise FALSE

---

is_rgo_model *Check Whether Input is an 'rgo_model' Object*

Description

Check Whether Input is an 'rgo_model' Object

Usage

```
is_rgo_model(x)
```

Arguments

x	R object to be tested

Value

TRUE if x inherits "rgo_model", otherwise FALSE

is_stat *Test Whether Object is an* rgo_stat

Description

Test Whether Object is an rgo_stat

Usage

```
is_stat(x)
```

Arguments

x	object to be tested

Value

TRUE or FALSE

is_test *Test Whether Object is an* rgo_test

Description

Test Whether Object is an rgo_test

Usage

```
is_test(x)
```

Arguments

x	object to be tested

Value

TRUE or FALSE

length_models *Calculate the Number of Models in an* rgo *Object*

Description

Calculate the Number of Models in an rgo Object

Usage

```
length_models(x)
```

Arguments

x	object of class rgo to be searched

Value

R object

okun_state *Data for Estimating Okun's Law Across States*

Description

Unemployment rate and real gross state product data in the United States from 2018 through 2023. Data was downloaded from the Federal Reserve Economic Database (FRED) using fred_state from the eFRED package. Unemployment rates are sourced from the U.S. Bureau of Labor Statistics and are associated with FRED code "[state prefix]UR". GSP data is sourced from the U.S. Bureau of Economic Analysis and are associated with FRED code "[state prefix]RGSP".

Usage

```
okun_state
```

Format

A data frame with 306 rows and 4 columns:

year year of the observation, 2018 - 2023

state character vector containing the two-letter state code

ur average annual unemployment rate (%)

gsp annualized growth rate (%) of real gross state domestic product

Source

<https://fred.stlouisfed.org/>

ols *Ordinary Least Squares Regression*

Description

Fit a linear model using the regg ecosystem; equivalent to lm.

Usage

```
ols(x, ...)
```

Arguments

x	a data set, rgo_model_matrix, or regg object
...	If unnamed, the description of the model terms using reg_select. If named, arguments passed to regg. See 'details' below.

Value

object of class 'regg'

regg *Create a New Regression Model*

Description

Master regression modeling function to be called by regression methods.

Usage

```
regg(x, ...)
```

Arguments

x	Object passed to regg and upon which the methods are dispatched.
...	Other arguments passed on to methods or quoted and stored in the returned environment..

Details

A regg object is an environment with two classes: regg and rgo *(inherited by almost all objects created in the regg package)*. It contains two primary fields: "data", containing the location to the underlying data set used by the model, and "models", a list of rgo_model objects. Each rgo_model represents an individual regression model with its own terms, coefficients, fitting method, etc.

Most of the named fields passed to ... are folded into the rgo_model objects, which are chained environments that inherit from the previous model or regg object. This allows methods, data, and fitting parameters to be inherited across regression models. Note that these fields are not evaluated at the time of being added to the environment. If these fields need to be accessed, they should be evaluated first via reg_eval.

Any unnamed field in ... is treated as a regression term where the first object is the response variable by default and the rest represent independent variables. These objects are evaluated using reg_select.

The most important field passed to regg is method. This contains the method of fitting the regression model. The method should be a function that evaluates an rgo_model object, adds a "coefficient" field, and returns the object. Note that the model is only fit when the object is activated through printing, accessing model components, or other objects that explicitly need to fitted regression model. See fit for more details about the fitting process. Elements of the model, such as the model matrix or regression weights, can be accessed without fitting the model through functions such as get_x and get_y. If not method is supplied, ols will be used.

regg accepts three important fields by the user - se, stats, and tests. se is a function describing the method of calculating the regression standard errors. stats is a list of functions used to estimate regression statistics such as the R-squared or AIC. If a character is used instead of a function, it will be prefaced by "stat_" and the search path will be examined for a function of the relevant name. For example, "r2" will link to the function stat_r2. The tests field is also a list of function or character values, but "t" would link to the function test_t.

When developing a method that uses regg, you can use default_se, default_stats, and default_tests to set standard statistics and so forth that you wish to be estimated each time the method is used to fit the regression model.

Other notable fields include "label" which is for the user to label the model in the printed results, and standard fields in lm such as "weights", "subset", or "na.action".

Value

environment of class regg

---

reg_eval *Evaluate a Expression in* regg

Description

This function is intended for developers to evaluate quoted expressions passed to regg using reg_select semantics.

Usage

```
reg_eval(x, where, ...)
```

Arguments

x	expression to be evaluated
where	rgo object
...	objects passed to reg_select

Value

R object

reg_select tidyselect *for Regression Models*

Description

Select and convert variables in a data frame to a model matrix for use in a regression, using a mini-language that is mostly compatible with select from the tidyverse. reg_select allows for mathematical operations inside of the selection and includes more functions.

Usage

```
reg_select(
  data,
  ...,
  drop = TRUE,
  intercept = TRUE,
  intercept_name = "(Intercept)",
  model = NULL,
  numeric = TRUE,
  overwrite_names = TRUE,
  quoted = NULL,
  response = TRUE
)
```

Arguments

data	data.frame or similar object containing columns to be selected
...	columns to be selected. See details below.
drop	If TRUE, the result is coerced to the lowest possible dimension. Only works if numeric is FALSE.
intercept	should an intercept be included in the result? An intercept is only included if numeric is TRUE.
intercept_name	character providing name of the intercept in the return value
model	regression model from which information about the regression can be drawn
numeric	should character & factor vectors be converted into a matrix with values of 0 or 1? If TRUE, a matrix will be returned, otherwise a data.frame.
overwrite_names	should variable names include functions/operations used
quoted	optional quoted list to be used in place of ...
response	logical value specifying whether or not the first argument should be interpreted as the dependent variable. Alternatively, a numeric vector can be supplied to specify which variables, by position, are the dependent variables.

Details

reg_select is similar to a cross between dplyr's select and transmute. It allows the user to select columns from the supplied data set and perform mathematical operations, such as calculating the logarithm of multiple columns at once, or multiplying two columns together. Furthermore,

character and factor variables are automatically converted to a matrix of dummy variables, as long as dummy = TRUE *(default)* in the function call.

Most of the selection helpers from select are available in reg_select. For example, the following tidyselect helpers can be used inside of reg_select.

- - Remove variables from the selection.
- : Selects a range of consecutive variables.
- c() Combines selections.
- everything() Matches all variables.
- last_col() Selects the last variable, possibly with an offset.
- group_cols() Selects all grouping columns.
- starts_with() Selects all column whose name starts with the supplied prefix.
- ends_with() Selects all columns whose name ends with the supplied suffix.
- contains() Find which column names contain the literal string.
- matches() Matches all column names via the supplied regular expression.
- num_range() Matches a numerical range like x01, x02, x03.
- all_of() Matches variable names in a character vector. All names must be present, otherwise an error is thrown.
- any_of() Same as all_of(), except that no error is thrown for names that don't exist.
- where() Applies a function to all variables and selects those for which the function returns TRUE.

Notice that !, &, and | are not available. This is a compromise to allow operations to be performed on variables during the selection process. Instead, use the following functions.

- complement() Finds the complement of a set of variables. (!)
- intersect() Finds the intersection of two sets of variables, or the variables that are included in both. (&)
- union() Finds the union of two sets of variables. (|)

reg_select first searches the supplied data for any column names associated with symbols passed in .... Symbol names that are prefaced with two dots, .., will search the parent environment rather than the data set. Alternatively, anything wrapped by the var() function will be evaluated in the parent environment first before proceeding.

Note that mathematical functions can be applied to the output of the helper functions. For example, log(starts_with("s")) would find all columns whose name begins with "s", then takes the logarithm. Note that functions from the Math and Complex groups will not be applied to dummy variables created from character vectors. However, functions from the Ops and Summary groups do apply to dummy variables.

Many window functions from dplyr - such as lag and lead - are available to use within ggr_select. These functions should operate correctly with grouped data.

New functions that are useful for regression analysis are also available.

- ar() Include the specified lags of the dependent variable(s). use_y in ggr_select must not be FALSE.
- current() Access the current rgo_model.
- previous() Access the previous rgo_model. This may be chained to access even earlier models.

- trend() Create a trend variable.
- y() Access the dependent variable of the current `rgo_model`.

In addition, `:=` can be used to give variables a new name. If the object on the left-hand side of `:=` is a character or symbol, it will become the new name of the object on the right-hand side. If the object on the right is a matrix, then all instances of the old name in the columns will be replaced with the name on the left. If a function, or object convertable to a function via `as_lambda`, is on the left-hand side, then it will be applied to the original name and all column names.

Other functions may also be used to control the output and printing behavior of the resulting model.

- hide() Columns are included in the output, but are not shown in regression results.
- temp() Columns are not included in the returned output.

Value

'rgo_modelmatrix' object

`reg_step` *Sequentially Add X Variables to Regression*

Description

Adds a series of regression models to `x`, by sequentially adding the unnamed objects passed to the function.

Usage

```
reg_step(x, ...)
```

Arguments

`x`	`rgo` object containing a regression model.
`...`	If unnamed, variables to be added as independent variables in the new regression using `reg_select`. If named, additional fields passed to `regg`.

Value

environment of class `regg`

se_default *Calculate Default Standard Errors*

Description

Default function for estimating standard errors, as calculated from the QR matrix, if available.

Usage

```
se_default(x, ...)
```

Arguments

x	object of class rgo_model or regg from which standard errors are calculated.
...	Other arguments passed on to methods

Value

environment of class regg or rgo_model

stats_common *Calculate and Add Common Regression Statistics*

Description

Common statistics include the number of observations (N), the R-squared (R2) and adjusted R-squared (adj_R2). Each are included in the "stats" field of each rgo_model.

Usage

```
stat_adj_r2(x, ...)

stat_conf(x, ...)

stat_mae(x, ...)

stat_n(x, ...)

stat_r2(x, ...)

stat_rmse(x, ...)
```

Arguments

x	regg or rgo_model object
...	objects passed to methods

Value

numeric value of statistic

test_f *Perform F-Test on Regression*

Description

Estimates the F-statistic and p-values associated with the hypothesis test that all coefficients are equal to 0, unless otherwise specified.

Usage

```
test_f(x, ...)
```

Arguments

x	object of class rgo_model or regg.
...	Other arguments passed on to methods

Value

environment of class regg or rgo_model

test_t *Perform t-Test on Regression Coefficients*

Description

Estimates the t-statistic and p-values associated with the hypothesis test that each coefficient is equal to 0.

Usage

```
test_t(x, ...)
```

Arguments

x	object of class rgo_model or regg from which standard errors are calculated.
...	Other arguments passed on to methods

Value

environment of class regg or rgo_model

tidy *Methods for Augmenting Data*

Description

These methods extract information about the regression object and place into a tibble. tidy extracts the regression coefficient table, along with model names if appropriate. Each column glance represents a regression statistic such as the R-squared. augment adds information related to residuals, fitted values, and so forth to the data set.

Usage

```
## S3 method for class 'rgo_m'
augment(x, ...)

## S3 method for class 'rgo_model'
components(
  x,
  label = NULL,
  bind = TRUE,
  objects = c("resids", "fitted", "hat", "cooks"),
  ...
)

## S3 method for class 'regg'
components(x, bind = TRUE, ...)

## S3 method for class 'rgo_model'
glance(x, label = NULL, bind = TRUE, ...)

## S3 method for class 'regg'
glance(x, bind = TRUE, ...)

## S3 method for class 'rgo_model'
tidy(
  x,
  label = NULL,
  bind = TRUE,
  type = c("coef", "test", "stat", "full"),
  add_type = FALSE,
  ...
)

## S3 method for class 'regg'
tidy(x, bind = TRUE, ...)
```

Arguments

x	object of class regg or rgo_model
...	objects passed to methods

label	default label of the model when multiple models are present.
bind	should the results be bound into a single tibble data.frame?
type	character vector describing the type of results to be returned - coefficient table or tests.

Value

tibble object

---

use_stars *Use Specified Significance Indicators*

Description

Specify the noted significance levels in regression output and how they are marked.

Usage

```
use_stars(x, ..., list = NULL)

stars_default()
```

Arguments

x	object of class rgo
...	numeric values representing the noted significance levels, named with the desired mark.
list	named list or numeric vector. Used as an alternative to .... Alternatively, a function that accepts a vector of numbers and returns a character vector.

Value

the object x

Functions

- stars_default(): the default significance levels and markings in R, to be added to list

window_functions *Window Functions for* reg_select

Description

A collection of functions to be used in reg_select. The default, if used outside of reg_select, is the same as their dplyr equivalent.

Usage

```
lag(x, n = 1L, ...)

lead(x, n = 1L, ...)
```

Arguments

x	R object
n	integer describing the number of observations to lead or lag
...	objects passed to methods

Value

object with same type as x

Index