

Package: tvCoef (via r-universe)

November 1, 2024

Type Package

Title Linear Time-Varying Coefficient Models

Version 0.2.1

Description Convert linear model to a time-varying coefficient model using stepwise regressions, local regressions or state-space models.

License EUPL

Depends R (>= 3.6.0)

Imports dynlm, strucchange, tvReg, rjd3toolkit, rjd3sts (>= 2.0.0)

SystemRequirements Java (>= 17)

Remotes github::rjdverse/rjd3toolkit, github::rjdverse/rjd3sts

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Roxygen list(markdown = TRUE)

Repository <https://aqlt.r-universe.dev>

RemoteUrl <https://github.com/InseeFrLab/tvCoef>

RemoteRef HEAD

RemoteSha 080034a4c95544f9613e65dc771bafa0e1db7e65

Contents

bp_lm	2
break_data	3
full_exogeneous_matrix	3
gdp	4
get_data	4
get_formula	5
get_rmse	6
hansen_table	7

hansen_test	7
has_intercept	8
last_coef	8
lm_fenetre_fixe	9
lm_fixed_coeff	9
lm_residual_effect	10
manufacturing	10
moving_coefficients	12
oos_prev	13
piece_reg	14
resid_lm_fixed	15
rmse	16
rmse_prev	16
ssm_lm	17
ssm_lm_oos	19

Index	20
--------------	-----------

bp_lm	<i>Pie regression</i>
-------	-----------------------

Description

Computes as many regressions as breakup dates within a global model

Usage

```
bp_lm(x, left = TRUE, break_dates, tvlm = FALSE, ...)
```

Arguments

x	lm object. It is the global regression model.
left	logical. By default set to TRUE, i.e. the breakdate is the end date of each submodel
break_dates	optional, to indicate the breakup dates if they are known.
tvlm	By default set to FALSE. Indicates which model will be run on each sub data. FALSE means a lm will be run.
...	other arguments passed to <code>tvReg::tvLM()</code> .

Value

Returns an element of class bp_lm. It is a list containing the following elements:

model	all computed models, each of class lm or tvlm according to the parameter specified above
start	start date of the time serie
end	end date of the time serie

frequency	frequency of the time serie
breakdates	a list of the breakup dates
left	same as the parameter specified above
tv1m	same as the parameter specified above

break_data	<i>Break data</i>
------------	-------------------

Description

Splits a database according to one (or more) date

Usage

```
break_data(x, break_dates, left = TRUE, names = NULL, ...)
```

Arguments

x	a ts or mts object to split
break_dates	the date(s) at which you want to divide the data
left	logical. By default set to TRUE, i.e. the breakdate is the end date of each subcolumn
names	optional vector containing the names of the variables used to build the splitted data. By default the function try to guess the names from the x parameter.
...	other unused arguments

Value

a mts containing as many times more data columns than breakdates

full_exogeneous_matrix	<i>Extract Full Transformed Exogoneous matrix</i>
------------------------	---

Description

Extract Full Transformed Exogoneous matrix

Usage

```
full_exogeneous_matrix(model, ...)
```

Arguments

model	the model
...	other unused parameters

gdp	<i>French GDP</i>
-----	-------------------

Description

Dataset containing the quarterly growth of the total gross domestic product (GDP) of France and quarterly series of the French business climate in level and in difference.

Usage

gdp

Format

A quarterly ts object from 1949Q2 to 2024Q1.

Details

Dataset containing the quarterly growth of the total gross domestic product (GDP, "growth_gdp") of France, in volumes chained at previous year prices, seasonally and working day adjusted; and the French business climate in level.

The French business climate is a monthly series, it is transformed into three quarterly series using the month's place in the quarter. For example, "bc_fr_m1" contains the values in the first month of each quarter, and the "diff_fr_m1" is the difference of the previous variable (the 2000Q1 value corresponds to the difference in business climate between January 2000 and October 1999).

Data were downloaded March 15, 2024 and might therefore differ from the latest available data.

Source

INSEE

get_data	<i>Get data function</i>
----------	--------------------------

Description

Retrieves the data used in the model

Usage

```

get_data(model, ...)

## S3 method for class 'lm'
get_data(model, start = 1, frequency = 1, ...)

## S3 method for class 'dynlm'
get_data(model, ...)

## S3 method for class 'tv1m'
get_data(model, end = numeric(), frequency = 1, ...)

## S3 method for class 'bp_lm'
get_data(model, ...)

## S3 method for class 'piece_reg'
get_data(model, ...)

```

Arguments

model	the model
...	other unused parameters.
start	the start of the data.
frequency	the frequency of the data.
end	the end of the data.

get_formula	<i>Extract Formula From model</i>
-------------	-----------------------------------

Description

Extract Formula From model

Usage

```

get_formula(x)

## Default S3 method:
get_formula(x)

```

Arguments

x	the model.
---	------------

get_rmse	<i>Get elements of rmse_prev</i>
----------	----------------------------------

Description

Get elements of `rmse_prev`

Usage

```
get_lm_coef(model)
get_tvlm_coef(model, date, variable)
get_tvlm_bw(model)
get_rmse_bw_small(model, ...)
```

Arguments

<code>model</code>	result of rmse_prev fonction
<code>date</code>	the date on which we want the tvlm coefficients
<code>variable</code>	integer. The variable of which we want the coefficients at the given date
<code>...</code>	other unused parameters

Details

`get_lm_coef` allows to get all coefficients of all linear regression prediction models

`get_tvlm_coef` allows to get all coefficients of the variable at a certain date of all local regression prediction models

`get_tvlm_bw` allows to get the bandwidth of all local regression prediction models

`get_rmse_bw_small` allows to get rmse of linear, local and piecewise regression prediction models, when the bandwidth of the prediction model is different from 20.

`get_coef_plot` plot `get_tvlm_coef` of a certain variable at a certain date, `get_lm_coef` of the same variable and the bandwidth of all prediction models, thanks to `get_tvlm_bw`. It also highlights when the bandwidth is equal to 20.

hansen_table	<i>Hansen Table</i>
--------------	---------------------

Description

Hansen Table

Usage

```
hansen_table
```

Format

An object of class `data.frame` with 20 rows and 7 columns.

Source

Hansen, Bruce E. 1990. "Lagrange multiplier tests for parameter instability in non-linear models". *University of Rochester*. <https://users.ssc.wisc.edu/~bhansen/papers/LMTests.pdf>.

hansen_test	<i>Hansen Test</i>
-------------	--------------------

Description

Performs Hansen test

Usage

```
hansen_test(x, var, sigma = FALSE)
```

Arguments

x	lm object.
var	variables used for the joint test. By default all the variable are used.
sigma	logical indicating if the joint test should include the variance

Details

Perform Hansen test, which indicates if the variance of a model, a global model and the coefficients of the variable within this model are likely to be unstable over time.

HO: the coefficient/model is stable over time.

References

Bruce E Hansen "Testing for parameter instability in linear models". Journal of policy Modeling (1992)

Examples

```
model_gdp <- lm(
  formula = growth_gdp ~ bc_fr_m1 + diff_bc_fr_m1,
  data = gdp
)
hansen_test(model_gdp)
```

has_intercept	<i>Check if model has intercept</i>
---------------	-------------------------------------

Description

Check if model has intercept

Usage

```
has_intercept(x)
```

Arguments

x	a model
---	---------

last_coef	<i>Extract Last Coefficients</i>
-----------	----------------------------------

Description

Get last coefficients of lm or tvLM models.

Usage

```
last_coef(x)
```

Arguments

x	a 'tvlm' or 'lm' object
---	-------------------------

lm_fenetre_fixe	<i>Fixed Window Regression</i>
-----------------	--------------------------------

Description

Fixed Window Regression

Usage

```
lm_fenetre_fixe(formula, data, nbw = 1)
```

Arguments

formula	a formula object.
data	time series data.
nbw	number of windows.

Value

Return an object of class "lmffixe". Return all models, from which we can extract the usual coefficients, residuals, and fitted.values. And the divisor chosen by the function (arbitrary the middle one), the period, i.e. the length of each sub models, and the frequency of the data.

lm_fixed_coeff	<i>Fixed and variable coefficients regressions</i>
----------------	--

Description

Computes different types of regressions with some coefficients fixed and others allowed to vary

Usage

```
lm_fixed_coeff(formula, data, fixed_var, ...)
```

Arguments

formula	a formula object.
data	time series data.
fixed_var	chosen variables whose coefficients aren't allowed to vary through time
...	further arguments passed to tvReg::tvLM()

Value

global_model	the simple lm model
linear_reg	simple lm model with fixed coefficients
piecewise_reg	bp_lm model with fixed coefficients
tv_reg	tv_lm model with fixed coefficients

lm_residual_effect	<i>Residual Effect Regression</i>
--------------------	-----------------------------------

Description

Checks there is no residual effect in a model

Usage

```
lm_residual_effect(x, var = c(-1))
```

Arguments

x	lm object
var	the variables on which the residuals are to be regressed. By default use them all and cancel the explained variable

Examples

```
data("AirPassengers")
model <- lm(AirPassengers ~ time(AirPassengers))
lm_residual_effect(model)
```

manufacturing	<i>Business Surveys</i>
---------------	-------------------------

Description

Dataset containing the quarterly growth of production in the manufacturing sector and its main sub-sectors, quarterly balance of opinion of business surveys published by INSEE and Banque de France and quarterly overhang of the industrial production index.

Usage

```
manufacturing
```

Format

A quarterly ts object from 1949Q2 to 2024Q1.

Details

Dataset containing the quarterly growth of production in the manufacturing sector and its main sub-sectors and quarterly series series of business surveys published by INSEE and Banque de France.

The sectors studied are:

- Manufacturing industry
- Food products and beverages (C1)
- Capital goods (C3)
- Transport equipments (C4)
- Other manufacturing (C5)

"manuf_prod" contains the quarterly growth in production in the manufacturing sector, and the sub-sectors are in the form "prod_c1", "prod_c3", "prod_c4" and "prod_c5".

The overhang of the industrial production index corresponds to the quarterly growth obtained extending the series by the last known value:

- "overhang_ipi0" is the quarterly growth obtained extending the series by the last value of the previous quarter (December, March, June, September);
- "overhang_ipi1" is the quarterly growth obtained extending the series by the first value of the current quarter (January, April, July, October);
- "overhang_ipi2" is the quarterly growth obtained extending the series by the second value of the current quarter (February, May, August, November).

The business surveys being monthly, the balance of opinion are transformed into three quarterly series using the month's place in the quarter (for example taking the values of January, April, July and October). Variable names are constructed as the combination of several codes defined as follows:

- data source code (INSEE, ins, or Banque de France, bdf);
- name of the balance of opinion:

Code	Definition
bc	Business climate
oscd	Overall order books
tppa and prodpas	Past production
tppre and prodpre	Personal production expectation
sitcar	Situation of order books
evocar	Evolution of order books
prix	Selling prices
stocks	Inventories of finished goods
tres	Cash position
tuc	Capacity utilisation rate

- sector: nothing for the manufacturing industries and "c1", "c3", "c4" or "c5" for the sub-sectors;
- place of the month in the quarter: m1, m2 or m3 for the first, second or third month of the quarter.

The dataset also contains some dummies labelled "indYYYYQX", where YYYY is the year and X is the quarter.

Source

INSEE, Banque de France

moving_coefficients *Detect Fixed or Moving Coefficients*

Description

Functions to test if any coefficient is fixed or moving according to the Hansen test ([hansen_test\(\)](#))

Usage

```
moving_coefficients(
  x,
  a = c(5, 1, 2.5, 7.5, 10, 20),
  sigma = FALSE,
  intercept = TRUE
)

fixed_coefficients(
  x,
  a = c(5, 1, 2.5, 7.5, 10, 20),
  sigma = FALSE,
  intercept = TRUE
)
```

Arguments

x	lm object.
a	level
sigma	logical indicating if the joint test should include the variance
intercept	boolean indicating if the intercept should be considered as a moving coefficient when at least one other variable is moving.

Value

NULL if no variable selected, otherwise the order of the variables.

oos_prev	<i>Out of sample forecast (or simulated out of sample)</i>
----------	--

Description

Out of sample forecast (or simulated out of sample)

Usage

```
oos_prev(model, date = 28, period = 1, ...)

## S3 method for class 'lm'
oos_prev(model, date = 28, period = 1, data = NULL, ...)

## S3 method for class 'piece_reg'
oos_prev(model, date = 28, period = 1, ...)

## S3 method for class 'tv1m'
oos_prev(
  model,
  date = 28,
  period = 1,
  data_est = NULL,
  fixed_bw = FALSE,
  bw = NULL,
  end = numeric(),
  frequency = 1,
  ...
)

## S3 method for class 'bp_lm'
oos_prev(
  model,
  date = 28,
  period = 1,
  data_est = NULL,
  data,
  fixed_bw = FALSE,
  bw = NULL,
  ...
)

## S3 method for class 'piece_reg'
oos_prev(model, date = 28, period = 1, ...)
```

Arguments

model an object used to select a method

date	choose when we want to start the revision process after the start date. By default set to 28 periods.
period	choose by how many values we want to move forward. By default set to 1.
...	other arguments
data	a ts object containing the variables in the model. Necessary only when x is a formula.
data_est, end, frequency	optional arguments to specify the data used to estimate the model, the last date and the frequency
fixed_bw	logical, by default set to FALSE. Indicates if the bandwidth has to be computed again in the forecast model, or if it takes the value of the bandwidth of the tvlm model.
bw	bandwidth of the local regression (when tvlm = TRUE).

Value

oos_prev returns an object of class revision, only for models of class `lm` and `tvlm`. For an object of class `bplm` it returns the same forecasts and residuals as below. An object of class revision is a list containing the following elements:

model	all models used to forecast
debut	same as date chosen earlier
intervalle	same as period chosen earlier
end_dates	a vector of all end date of each models
frequency	the frequency of the data
forecast	the forecast
residuals	the errors of the forecast

Examples

```
data_gdp <- window(gdp, start = 1980, end = c(2019, 4))
reg_lin <- lm(
  formula = growth_gdp ~ bc_fr_m1 + diff_bc_fr_m1,
  data = data_gdp
)
oos <- oos_prev(reg_lin)
```

piece_reg

Piecewise regression

Description

Computes one global linear regression, on splitted data

Usage

```

piece_reg(
  x,
  break_dates = NULL,
  fixed_var = NULL,
  tvlm = FALSE,
  bw = NULL,
  left = TRUE,
  ...
)

```

Arguments

<code>x</code>	lm object. It is the global regression model
<code>break_dates</code>	optional, to indicate the breakdates if they are known. By default set to NULL.
<code>fixed_var</code>	fixed variables (not splitted using <code>break_dates</code>).
<code>tvlm</code>	By default set to FALSE. Indicates which model will be run on each sub data. FALSE means a <code>lm</code> will be run.
<code>bw</code>	bandwidth of the local regression (when <code>tvlm = TRUE</code>).
<code>left</code>	logical. By default set to TRUE, i.e. the breakdate is the end date of each submodel
<code>...</code>	other arguments passed to <code>tvReg::tvLM()</code> .

Details

Computes possible breakdates if not filled in. Uses function `break_data` and run a linear regression on the same splitted data.

Value

Returns an element of class `lm`

<code>resid_lm_fixed</code>	<i>Extract data frame for <code>lm_fixed_coeff</code></i>
-----------------------------	---

Description

According to parameter `fixed_var`, computes a new explained variable, which is the explained variable minus the product between estimated coefficients and values of the fixed variables.

Usage

```
resid_lm_fixed(x, fixed_var)
```

Arguments

x lm model
 fixed_var list of variables that don't vary through time according to [hansen_test](#)

Value

A new environment where the explained variable is named "fixed".

rmse	<i>Root mean squarred error</i>
------	---------------------------------

Description

Root mean squarred error

Usage

```
rmse(resid)
```

Arguments

resid the residuals vector on which rmse will be calculated

rmse_prev	<i>Computes RMSE of different models</i>
-----------	--

Description

Computes 6 models: linear regression, piecewise regression (with linear regression and local regression), local regression (with [tvLM](#)), piecewise regression with some fixed coefficients and tvlm with fixed coefficients. Computes 6 rmse on their residuals and 6 others on the residuals of the predictions of these models.

Usage

```
rmse_prev(x, data, fixed_var = NULL, fixed_bw = FALSE, ...)
```

```
## S3 method for class 'formula'
```

```
rmse_prev(x, data, fixed_var = NULL, fixed_bw = FALSE, ...)
```

```
## S3 method for class 'lm'
```

```
rmse_prev(x, data, fixed_var = NULL, fixed_bw = FALSE, ...)
```


Arguments

x	an object of class formula or lm, that is the description of the model to be fitted, or the model itself.
data	a ts object containing the variables in the model. Necessary only when x is a formula.
fixed_var	which variables of the model should have their coefficients fixed for the models with fixed coefficients. Obtained thanks to the hansen test.
fixed_bw	logical, by default set to FALSE. Indicates if the bandwidth has to be computed again in the forecast model, or if it takes the value of the bandwidth of the tvlm model.
...	additional arguments

Details

In additional arguments, parameters as date and period can be informed. As in [oos_prev](#) they are by default respectively set to 28 and 1.

To estimate forecast models, the function [oos_prev](#) is used.

For the forecasts of the two models with fixed coefficients, fixed coefficients are re-estimated at each date, before [bp_lm](#) or [tvLM](#) are run on moving variables.

Value

Returns an object of class prev which is a list containing the following elements:

model	a list of the 6 explanatory models
forecast	a list of the 6 predictions of the 6 previous models
rmse	a list of the 2 computed rmse, in sample and out of sample

ssm_lm *State space model*

Description

Computes state space model with one equation. Starting with a simple lm model, build the all state space model and run it.

Use [rjd3sts](#) packages.

Usage

```
ssm_lm(
  x,
  trend = FALSE,
  var_intercept = 0,
  var_slope = 0,
```

```

var_variables = 0,
fixed_var_intercept = TRUE,
fixed_var_trend = TRUE,
fixed_var_variables = TRUE,
...,
remove_last_dummies = FALSE,
intercept = TRUE
)

```

Arguments

`x` a `lm` or `dynlm` model

`trend` boolean indicating if the model should have a trend.

`var_intercept`, `var_slope` variance of the intercept (used if `intercept = TRUE`) and the slope (used if `trend = TRUE`).

`var_variables` variance of the other variables: can be either a single value (same variance for all the variables) or a vector specifying each variance.

`fixed_var_intercept`, `fixed_var_trend`, `fixed_var_variables` logical indicating if the variance are fixed or estimated.

... other arguments used in `rjd3sts::estimate()`.

`remove_last_dummies` boolean indicating if current dummies (i.e.: only 0 and 1 at the last date) should be removed.

`intercept` boolean indicating if the model should have an intercept.

Value

Returns a list containing:

`smoothed_states` $E[a_t|y_0, \dots, y_n]$

`smoothed_stdev` $\sqrt{V[a_t|y_0, \dots, y_n]}$

`filtering_states` $E[a_t|y_0, \dots, y_{t-1}]$

`filtering_stdev` $\sqrt{V[a_t|y_0, \dots, y_{t-1}]}$

`parameters` some estimation parameters

`data` data used in the original model

Examples

```

data_gdp <- window(gdp, start = 1980, end = c(2019, 4))
reg_lin <- lm(
  formula = growth_gdp ~ bc_fr_m1 + diff_bc_fr_m1,
  data = data_gdp
)

```

```
ssm <- ssm_lm(reg_lin, fixed_var_intercept = FALSE, fixed_var_variables = FALSE)
ssm
summary(ssm)
```

ssm_lm_oos

Out of sample forecast of state space model

Description

Computes out of sample forecasts of a given state space model. Unlike [ssm_lm](#) it can manage dummies.

Usage

```
ssm_lm_oos(
  x,
  trend = FALSE,
  var_intercept = 0,
  var_slope = 0,
  var_variables = 0,
  fixed_var_intercept = TRUE,
  fixed_var_trend = TRUE,
  fixed_var_variables = TRUE,
  date = 28,
  ...
)
```

Arguments

x	a lm or dynlm model
trend	boolean indicating if the model should have a trend.
var_intercept, var_slope	variance of the intercept (used if <code>intercept = TRUE</code>) and the slope (used if <code>trend = TRUE</code>).
var_variables	variance of the other variables: can be either a single value (same variance for all the variables) or a vector specifying each variance.
fixed_var_intercept, fixed_var_trend, fixed_var_variables	logical indicating if the variance are fixed or estimated.
date	choose when we want to start the revision process after the start date. By default set to 28 periods.
...	other arguments used in <code>rjd3sts::estimate()</code> .

Value

Returns all coefficients of all variables and the residual

Index

- * **datasets**
 - gdp, 4
 - hansen_table, 7
 - manufacturing, 10
- bp_lm, 2, 17
- break_data, 3, 15
- fixed_coefficients
 - (moving_coefficients), 12
- full_exogeneous_matrix, 3
- gdp, 4
- get_data, 4
- get_formula, 5
- get_lm_coef (get_rmse), 6
- get_rmse, 6
- get_rmse_bw_small (get_rmse), 6
- get_tvlm_bw (get_rmse), 6
- get_tvlm_coef (get_rmse), 6
- hansen_table, 7
- hansen_test, 7, 16
- hansen_test(), 12
- has_intercept, 8
- last_coef, 8
- lm, 2, 14, 15
- lm_fenetre_fixe, 9
- lm_fixed_coeff, 9
- lm_residual_effect, 10
- manufacturing, 10
- moving_coefficients, 12
- oos_prev, 13, 17
- piece_reg, 14
- resid_lm_fixed, 15
- rjd3sts::estimate(), 18, 19
- rmse, 16
- rmse_prev, 6, 16
- ssm_lm, 17, 19
- ssm_lm_oos, 19
- tvLM, 16, 17
- tvlm, 14
- tvReg::tvLM(), 2, 9, 15