

# Package ‘MMLR’

July 21, 2025

**Type** Package

**Title** Fitting Markov-Modulated Linear Regression Models

**Version** 0.2.0

**Maintainer** Nadezda Spiridovska <Spiridovska.N@tsi.lv>

**Description** A set of tools for fitting Markov-modulated linear regression, where responses  $Y(t)$  are time-additive, and model operates in the external environment, which is described as a continuous time Markov chain with finite state space.

Model is proposed by Alexander Andronov (2012) <[doi:10.48550/arXiv.1901.09600v1](https://doi.org/10.48550/arXiv.1901.09600v1)> and algorithm of parameters estimation is based on eigenvalues and eigenvectors decomposition. Markov-switching regression models have the same idea of varying the regression parameters randomly in accordance with external environment. The difference is that for Markov-modulated linear regression model the external environment is described as a continuous-time homogeneous irreducible Markov chain with known parameters while switching models consider Markov chain as unobserved and estimation procedure involves estimation of transition matrix.

These models have significant differences in terms of the analytical approach.

Also, package provides a set of data simulation tools for Markov-modulated linear regression (for academical/research purposes).

Research project No. 1.1.1.2/VIAA/1/16/075.

**License** GPL ( $\geq 2$ )

**Encoding** UTF-8

**LazyData** true

**Imports** stats, pracma

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Author** Nadezda Spiridovska [aut, cre],  
Diana Santalova [ctb]

**Repository** CRAN

**Date/Publication** 2020-01-09 12:30:11 UTC

## Contents

Aver_soj_time . . . . .	2
B_est . . . . .	3
randomizeInitState . . . . .	4
randomizeTau . . . . .	5
randomizeX . . . . .	6
VarY . . . . .	7
Xreg . . . . .	8
Ysimulation . . . . .	9
<b>Index</b>	<b>10</b>

---

Aver_soj_time	<i>Calculating the average sojourn time in each state</i>
---------------	---

---

### Description

Calculating expectation of sojourn times in states for the observed time and for given initial state, using eigenvalues and eigenvectors.

### Usage

```
Aver_soj_time(ii, tau_observed, Q)
```

### Arguments

ii	number (scalar)
tau_observed	number (scalar), observed time
Q	Matrix (m x m), m - number of states

### Details

Calculating expectation of sojourn times in states for the observed time (tau\_observed) and if initial state is given (ii). Matrix Q is so-called Generator matrix:  $Q = \lambda - \Lambda$ , where  $\lambda$  is matrix with known transition rates from state  $s_i$  to state  $s_j$ , and  $\Lambda$  is diagonal matrix with a vector  $(\Lambda_1, \dots, \Lambda_m)$  on the main diagonal, where m is a number of states of external environment. Eigenvalues and eigenvectors are used in calculations.

### Value

Vector of average sojourn times in each state. Vector components in total should give observation time (tau\_observed).

**Examples**

```
lambda <- matrix(c(0, 0.33, 0.45, 0), nrow = 2, ncol = 2, byrow = TRUE)
m <- nrow(lambda)
ld <- as.matrix(rowSums(lambda))
Lambda <- diag(as.vector(ld))
Generator <- t(lambda) - Lambda
Aver_soj_time(1,10,Generator)
```

---

B_est	<i>Estimation of unknown Markov-modulated linear regression model parameters using GLSM</i>
-------	---

---

**Description**

This function is used to fit Markov-modulated linear regression models with two states of external environment. This function estimates Markov-modulated linear regression model parameters, using GLSM. Function uses the algorithm based on eigenvalues and eigenvectors decompositions.

**Usage**

```
B_est(tGiven, initState, X, Y, lambda, W = TRUE)
```

**Arguments**

tGiven	Vector of the observed times (n x 1), n – number of observations
initState	Vector of the initial states (n x 1), n – number of observations
X	Matrix of predictors (n x k), n - number of observations, k - number of columns (k - 1 - number of regressors).
Y	Vector of the responses Y, n – number of observations
lambda	Matrix with the known transition rates $\lambda_{i,j}$ , (m x m), m – number of states
W	an optional logical variable indicating should vector of weights be used in the fitting process or not. If TRUE, matrix with weights is used (that is, inverse values to tGiven – observed times).

**Details**

Function calculates the following expression:  $\beta$  (Fig.2), where vector of average sojourn times in each state  $t_i$  is calculated using function Aver\_soj\_time,  $t_i$  is an element of tGiven,  $x_i$  is a vector of matrix X.

**Value**

Vector of estimated parameters  $\beta$

**Examples**

```
lambda <- matrix(c(0, 0.33, 0.45, 0), nrow = 2, ncol = 2, byrow = TRUE)
Xtest <- cbind(rep_len(1,10),c(2,5,7,3,1,1,2,2,3,6), c(5,9,1,2,3,2,3,5,2,2))
tGiven <- matrix (c(6,4.8,1,2.6,6.4,1.7,2.9,4.4,1.5,3.4), nrow = 10, ncol = 1)
Y <- matrix(c (5.7, 9.9, 7.8, 14.5, 8.2, 14.5, 32.2, 3.8, 16.5, 7.7),nrow = 10, ncol = 1)
initState <- matrix (c(2,1,1,2,2,2,1,1,2,1),nrow = 10, ncol = 1)
B_est(tGiven,initState,Xtest,Y,lambda,W = 1)
```

---

randomizeInitState      *Transformation of vector with initial states I for various observations.  
Data preparation stage for simulation.*

---

**Description**

Additional function to be used for simulation purposes (academical or research). Transforming of vector with initial states I for various observations with respect to stationary distribution of the states for the random environment.

**Usage**

```
randomizeInitState(StatPr, X, p = 1)
```

**Arguments**

StatPr	Vector (m x 1), m - number of states, m = 2,3,.. .The vector with stationary probabilities, user-defined vector.
X	Matrix (n x k), n - number of observations, k - number of columns (k - 1 - number of regressors). The matrix is needed to get the number of observations.
p	Scalar (from 1 to +inf), random number for simulation. The default value is 1.

**Details**

The initial states (m - number of states, m = 2,3,...) for various observations are independent and are chosen with respect to stationary distribution of the states for the random environment. The vector with stationary probabilities is user-defined vector.

**Value**

Vector with new initial states, according to stationary distribution of the states for the random environment.

**Examples**

```
Xtest <- cbind(rep_len(1,10),c(2,5,7,3,1,1,2,2,3,6), c(5,9,1,2,3,2,3,5,2,2))
StatPr <- matrix (c(0.364,0.242,0.394), nrow = 3, ncol = 1)
randomizeInitState(StatPr,Xtest,1)
```

---

randomizeTau	<i>Transformation of the observed time vector tau. Data preparation stage for simulation</i>
--------------	--

---

### Description

Additional function to be used for simulation purposes (academical or research). Transforming of the observed time vector tau according to user preferences. Random disturbances are entered in the initial values of the vector tau. The expectation of new observed times coincides with initial values of vector tau.

### Usage

```
randomizeTau(tau, p, k0 = 2, k1 = 1)
```

### Arguments

tau	Vector (n x 1), n - number of observations
p	Scalar (from 1 to +inf), random number for simulation. The default value is 1
k0	Scalar (from 1 to +inf). Multiplicative parameter for transforming the initial value The default is k0 = 2.
k1	Scalar. The number of digits after the comma when rounded. The default is 1.

### Details

Initial values of observation times are multiplied by a random value ( $\tau_{i,k} \times \text{rnd}(0, 1)$ ). All times are independent and time of *i*th observation has uniform distribution on  $(0, k\tau_{i,k})$ .

### Value

Vector with new observation times, according to user preferences.

### Examples

```
tGiven <- matrix (c(6,4.8,1,2.6,6.4,1.7,2.9,4.4,1.5,3.4), nrow = 10, ncol = 1)
randomizeTau(tGiven,1,2,2)
```

---

randomizeX	<i>Transformation of regressors' matrix X. Data preparation stage for simulation</i>
------------	--

---

### Description

Additional function to be used for simulation purposes (academical or research). Transforming the matrix of regressors according to user preferences. Random disturbances (according to a chosen distribution) are entered in the initial values of the matrix X. The expectation of the resulting matrix coincides with the initial matrix X.

### Usage

```
randomizeX(X, p = 1, k0 = 1, k1 = 0.5, k2 = 1, k3 = 1)
```

### Arguments

X	Matrix (n x k), n - number of observations, k - number of columns (k - 1 - number of regressors). Note, that 1st column contains only ones (1) (intercept)
p	Scalar (from 1 to +inf), random number for simulation. The default value is 1
k0	Scalar. Number from 1 to 3 (distribution selection). k0 = 1 - uniform distribution (RV $Z \sim U(K1, k2)$ ). k0 = 2 - exponential distribution (RV $Z \sim \exp(\lambda)$ ). k0 = 3 - Gamma distribution (RV $Z \sim \text{gamma}(\text{shape}, \text{rate})$ ). The default value is k0 = 1.
k1	Scalar. 1) If k0 = 1, then k1 is a left boundary of uniform distribution (RV $Z \sim U(k1, k2)$ ). 2) If k0 = 2, then k1 is a parameter lambda of exponential distribution (RV $Z \sim \exp(\lambda)$ ). 3) If k0 = 3, then k1 is a shape parameter of Gamma distribution (RV $Z \sim \text{gamma}(\text{shape}, \text{rate})$ ). The default is k1 = 0.5.
k2	Scalar. 1) If k0 = 1, then k2 is a right boundary of uniform distribution (RV $Z \sim U(k1, k2)$ ). 2) If k0 = 3, then k2 is a rate parameter of Gamma distribution (RV $Z \sim \text{gamma}(\text{shape}, \text{rate})$ ). The default is k2 = 1.
k3	Scalar. The number of digits after the comma when rounded. The default value is 1.

### Details

Random perturbations are added to the initial values of matrix X elements ( $X_{i,j} + \text{rnd}$ ), which are distributed according to a chosen distribution (possible alternatives: uniform, exponential and gamma distribution).

### Value

New transformed matrix of regressors (n x k), according to user preferences.

**Examples**

```
Xtest <- cbind(rep_len(1,10),c(2,5,7,3,1,1,2,2,3,6), c(5,9,1,2,3,2,3,5,2,2))
randomizeX(Xtest,1,1,1,2,2)
```

---

 VarY

*Estimation of the variance of the response Y*


---

**Description**

This function is used for calculation of the variance of the response Y (Var(Y))

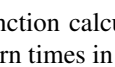
**Usage**

```
VarY(bb, sigma, i, x, tau, la)
```

**Arguments**

bb	Matrix (k x m), k - number of columns (k - 1 - number of regressors), m - number of states, m = 2,3,...
sigma	Scalar, the standard deviation of the disturbance term
i	number (scalar), initial state
x	Row-vector of the matrix of predictors X (1 x k), k - number of columns.
tau	number (scalar), observed time
la	Matrix with the known transition rates $\lambda_{i,j}$ , (m x m), m – number of states

**Details**

Function calculates the following expression:  "Fig.3", where vector of average sojourn times in each state  $t_{i,j}$  is calculated using function `Aver_soj_time`

**Value**

Estimation of the variance of the response Y, scalar

**Examples**

```
Xtest <- cbind(rep_len(1,10),c(2,5,7,3,1,1,2,2,3,6), c(5,4,1,2,3,2,3,5,2,2))
tGiven <- matrix (c(0.9,1.18,1,1.6,1.4,1.7,1.9,1.45,1.5,2.14), nrow = 10, ncol = 1)
initState <- matrix (c(2,1,1,2,2,2,1,1,2,1),nrow = 10, ncol = 1)
lambda <- matrix(c(0, 0.33, 0.45, 0), nrow = 2, ncol = 2, byrow = TRUE)
beta <- matrix(c(1, 2, 3, 4, 6, 8), nrow = 3, ncol = 2, byrow = TRUE)
VarY(beta,1,2,Xtest[3,],10,lambda)
```

Xreg

*Preparing data for parameter estimation procedure***Description**

Regressors matrix formation taking into account observation times and initial states. Kronecker product is used.

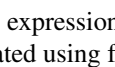
**Usage**

```
Xreg(tGiven, initState, X, lambda)
```

**Arguments**

tGiven	Vector of the observed times (n x 1), n – number of observations
initState	Vector of the initial states (n x 1), n – number of observations
X	Matrix of predictors (n x k), n - number of observations, k - number of columns (k - 1 - number of regressors).
lambda	Matrix with the known transition rates $\lambda_{i,j}$ , (m x m), m – number of states

**Details**

Function calculates the following expression  "Fig.1", where vector of average sojourn times in each state is calculated using function Aver\_soj\_time.

**Value**

Matrix (n x 2k)

**Examples**

```
Xtest <- cbind(rep_len(1,10),c(2,5,7,3,1,1,2,2,3,6), c(5,9,1,2,3,2,3,5,2,2))
tGiven <- matrix (c(6,4.8,1,2.6,6.4,1.7,2.9,4.4,1.5,3.4), nrow = 10, ncol = 1)
initState <- matrix (c(2,1,1,2,2,2,1,1,2,1),nrow = 10, ncol = 1)
lambda <- matrix(c(0, 0.33, 0.45, 0), nrow = 2, ncol = 2, byrow = TRUE)
Xreg(tGiven, initState, Xtest, lambda)
```



---

Ysimulation	<i>Simulation of the vector of responses Y. Data preparation stage for simulation.</i>
-------------	--

---

### Description

Additional function to be used for simulation purposes (academical or research). Simulating the vector of responses Y according to the formula (see details).

### Usage

```
Ysimulation(t, I, X, lambda, sigma = 1, beta)
```

### Arguments

t	Vector of the observed time (n x 1), n – number of observations
I	Vector of the initial states (n x 1), n – number of observations
X	Matrix of predictors (n x k), n - number of observations, k - number of columns (k - 1 - number of regressors).
lambda	Matrix with the known transition rates $\lambda_{i,j}$ , (m x m), m – number of states
sigma	Scalar, the standard deviation of the disturbance term
beta	Matrix (k x m), k - number of columns (k - 1 - number of regressors), m - number of states, m = 2,3,.. .

### Details

The i-th response  $Y_i$  is defined by the following formula:  $Y_i(t) = x_i \beta + Z_i \text{sqrt}(\sigma^2)$ ,  $i=1, \dots, n$ .  
The vector with stationary probabilities is user-defined vector.

### Value

Vector with new response values of vector Y (n x 1)

### Examples

```
Xtest <- cbind(rep_len(1,10),c(2,5,7,3,1,1,2,2,3,6), c(5,4,1,2,3,2,3,5,2,2))
tGiven <- matrix(c(0.9,1.18,1,1.6,1.4,1.7,1.9,1.45,1.5,2.14), nrow = 10, ncol = 1)
initState <- matrix(c(2,1,1,2,2,2,1,1,2,1),nrow = 10, ncol = 1)
lambda <- matrix(c(0, 0.33, 0.45, 0), nrow = 2, ncol = 2, byrow = TRUE)
beta <- matrix(c(1, 2, 3, 4, 6, 8), nrow = 3, ncol = 2, byrow = TRUE)
Ysimulation(tGiven,initState,Xtest,lambda,1,beta)
```

# Index

Aver\_soj\_time, [2](#)

B\_est, [3](#)

randomizeInitState, [4](#)

randomizeTau, [5](#)

randomizeX, [6](#)

VarY, [7](#)

Xreg, [8](#)

Ysimulation, [9](#)