Package ‘EWS’

February 24, 2021

Type Package
Title Early Warning System
Version 0.2.0
Author Jean-Baptiste Hasse [aut], Quentin Lajaunie [aut, cre]
Maintainer Quentin Lajaunie <quentin_lajaunie@hotmail.fr>
Description The purpose of Early Warning Systems (EWS) is to detect accurately the occurrence of a crisis, which is represented by a binary variable which takes the value of one when the event occurs, and the value of zero otherwise. EWS are a toolbox for policymakers to prevent or attenuate the impact of economic downturns. Modern EWS are based on the econometric framework of Kauppi and Saikkonen (2008) <doi:10.1162/rest.90.4.777>. Specifically, this framework includes four dichotomous models, relying on a logit approach to model the relationship between yield spreads and future recessions, controlling for recession risk factors. These models can be estimated in a univariate or a balanced panel framework as in Cand elo, Dumitrescu and Hurlin (2014) <doi:10.1016/j.ijforecast.2014.03.015>. This package provides both methods for estimating these models and a dataset covering 13 OECD countries over a period of 45 years. In addition, this package also provides methods for the analysis of the propagation mechanisms of an exogenous shock, as well as robust confidence intervals for these response functions using a block-bootstrap method as in Lajaunie (2021). This package constitutes a useful toolbox (data and functions) for scholars as well as policymakers.

Depends R (>= 2.10)
License GPL-3
Encoding UTF-8
LazyData true
Imports numDeriv
NeedsCompilation no
Repository CRAN
Date/Publication 2021-02-24 10:30:06 UTC

R topics documented:

BlockBootstrapp ......................................................... 2
**BlockBootstrapp**

This function enables the estimation of the block size for resampling. The size of the blocks is computed as in Hall, Horowitz and Jing (1995). Then, the function returns in a matrix the new resampled input variables. These variables are then used to determine the confidence intervals of the response functions proposed by Lajaunie (2021).

**Description**

This function enables the estimation of the block size for resampling. The size of the blocks is computed as in Hall, Horowitz and Jing (1995). Then, the function returns in a matrix the new resampled input variables. These variables are then used to determine the confidence intervals of the response functions proposed by Lajaunie (2021).

**Usage**

```r
BlockBootstrapp(Dicho_Y, Exp_X, Intercept, n_simul)
```

**Arguments**

- `Dicho_Y` Vector of the binary time series.
- `Exp_X` Vector or Matrix of explanatory time series.
- `Intercept` Boolean value: TRUE for an estimation with intercept, and FALSE otherwise.
- `n_simul` Numeric variable equal to the total number of replications.

**Value**

A matrix containing the replications of the new resampled input variables. The matrix contains \( n \times S \) columns, where \( n \) denotes the number of input variables, and \( S \) denotes the number of replications.

**Author(s)**

Jean-Baptiste Hasse and Quentin Lajaunie
# NOT RUN {
# Import data
data("data_USA")

# Data process
Var_Y <- as.vector(data_USA$NBER)
Var_X <- as.vector(data_USA$Spread)

# Resample
results <- BlockBootstrapp(Dicho_Y = Var_Y, Exp_X = Var_X, Intercept = TRUE, n_simul = 100)

# print results
results
#}

## data_panel

Historical data for 13 OECD countries

### Description

data_USA contains:
- OECD based Recession Indicators for 13 OECD countries from the Peak through the Trough from 1975:03 to 2019:05
- Yield Spread (10Years TB minus 3Months TB) for 13 OECD countries from 1975:03 to 2019:05

List of countries: Australia, Belgium, Canada, France, Germany, Italy, Japan, the Netherlands, New Zealand, Sweden, Switzerland, the United Kinkdom, the United States.

### Usage

data("data_panel")

### Format

A data frame with 6903 observations on the following 4 variables.

- **country**  List of countries.
- **Date**  Vector of dates.
YIESPR historical yield spread for the 13 OECD countries.
OECD Historical binary variable related to historical recessions for the 13 OECD countries.

Source
https://fred.stlouisfed.org/

Examples
data("data_panel")
head("data_panel")

data_USA  

Historical data for the United States

Description
data_USA contains:
- NBER based Recession Indicators for the United States from 1953:04 to 2020:01
- 10Years TB for the United States from 1953:04 to 2020:01
- 3Months TB for the United States from 1953:04 to 2020:01
- Yield Spread (10Years TB minus 3Months TB) for the United States from 1975:03 to 2019:05

Usage
data("data_USA")

Format
A data frame with 268 observations on the following 5 variables.

Date  Vector of dates.
X10Y  Historical 10 years Treasury bond.
X3M  Historical 3 months Treasury bond.
Spread  Historical yield spread.
NBER  Historical binary variable related to historical recessions.

Source
https://fred.stlouisfed.org/

Examples
data("data_USA")
head("data_USA")
Description

This function provides a method to compute the optimal AM (Accuracy Measure) criterion. As defined in Candelon, Dumitrescu and Hurlin (2012), this approach consists in aggregating the number of crisis and calm periods correctly identified by the EWS. The optimal cut-off maximizes the number of correctly identified periods.

Usage

EWS_AM_Criterion(Var_Proba, Dicho_Y, cutoff_interval)

Arguments

Var_Proba  Vector containing the estimated probabilities obtained with the Logistic Estimation function.
Dicho_Y    Vector of the binary time series.
cutoff_interval  Numeric variable between 0 and 1.

Value

A numeric variable containing the optimal cut-off that maximizes the higher proportion of calm and crisis periods correctly identified.

Author(s)

Jean-Baptiste Hasse and Quentin Lajaunie

References


Examples

# NOT RUN {
# Import data
data("data_USA")
# Data process
Var_Y <- as.vector(data_USA$NBER)
Var_X <- as.vector(data_USA$Spread)

# Estimate the logit regression
Logistic_results <- Logistic_Estimation(Dicho_Y = Var_Y, Exp_X = Var_X, Intercept = TRUE,
                                       Nb_Id = 1, Lag = 1, type_model = 4)

# Vector of probabilities
vector_proba <- as.vector(rep(0,length(Var_Y)-1))
vector_proba <- Logistic_results$prob

# Vector of binary variables
Lag <- 1
vector_binary <- as.vector(rep(0,length(Var_Y)-1))
vector_binary <- Var_Y[(1+Lag):length(Var_Y)]

# optimal cut-off that maximizes the AM criterion
results <- EWS_AM_Criterion(Var_Proba = vector_proba, Dicho_Y = vector_binary,
                            cutoff_interval = 0.0001)

# print results
results
#

describe

EWS_CSA_Criterion  

Description

This function provides a method to compute the optimal CSA (Credit-Scoring Approach) criterion. As defined in Candelon, Dumitrescu and Hurlin (2012), this approach consists of calculating the difference between the sensitivity and the specificity. Sensitivity represents the proportion of crisis periods correctly identified by the EWS. Specificity is the proportion of calm periods correctly identified by the EWS. The optimal cut-off minimizes the absolute value of this difference.

Usage

EWS_CSA_Criterion(Var_Proba, Dicho_Y, cutoff_interval)

Arguments

Var_Proba  Vector containing the estimated probabilities obtained with the Logistic Estimation function.
Dicho_Y    Vector of the binary time series.
cutoff_interval  Numeric variable between 0 and 1.
Value

A numeric variable containing the optimal cut-off that minimizes the absolute value of the difference between the sensitivity and the specificity.

Author(s)

Jean-Baptiste Hasse and Quentin Lajaunie

References


Examples

```R
# NOT RUN {

# Import data
data("data_USA")

# Data process
Var_Y <- as.vector(data_USA$NBER)
Var_X <- as.vector(data_USA$Spread)

# Estimate the logit regression
Logistic_results <- Logistic_Estimation(Dicho_Y = Var_Y, Exp_X = Var_X, Intercept = TRUE,
                                        Nb_Id = 1, Lag = 1, type_model = 4)

# Vector of probabilities
vector_proba <- as.vector(rep(0,length(Var_Y)-1))
vector_proba <- Logistic_results$prob

# Vector of binary variables
Lag <- 1
vector_binary <- as.vector(rep(0,length(Var_Y)-1))
vector_binary <- Var_Y[(1+Lag):length(Var_Y)]

# optimal cut-off that minimizes the CSA criterion
results <- EWS_CSA_Criterion(Var_Proba = vector_proba, Dicho_Y = vector_binary,
                              cutoff_interval = 0.0001)

# print results
results

#}
```
Description

This function provides a method to compute the optimal NSR (Noise to Signal Ratio) criterion proposed by Kaminsky, Lizondo and Reinhart (1998). As defined in Candelon, Dumitrescu and Hurlin (2012), the NSR represents the ratio of the false alarms (type II error) to the number of crises correctly identified by the EWS for a given cut-off. The optimal cut-off minimizes the NSR criterion.

Usage

EWS_NSR_Criterion(Var_Proba, Dicho_Y, cutoff_interval)

Arguments

Var_Proba Vector containing the estimated probabilities obtained with the Logistic Estimation function.
Dicho_Y Vector of the binary time series.
cutoff_interval Numeric variable between 0 and 1.

Value

A numeric variable containing the optimal cut-off that minimizes the NSR criterion.

Author(s)

Jean-Baptiste Hasse and Quentin Lajaunie

References


Examples

# NOT RUN {

# Import data
data("data_USA")

# Data process
Var_Y <- as.vector(data_USA$NBER)
Var_X <- as.vector(data_USA$Spread)

# Estimate the logit regression
Logistic_results <- Logistic_Estimation(Dicho_Y = Var_Y, Exp_X = Var_X, Intercept = TRUE,
                                      Nb_Id = 1, Lag = 1, type_model = 4)

# Vector of probabilities
vector_proba <- as.vector(rep(0,length(Var_Y)-1))
vector_proba <- Logistic_results$prob

# Vector of binary variables
Lag <- 1
vector_binary <- as.vector(rep(0,length(Var_Y)-1))
vector_binary <- Var_Y[(1+Lag):length(Var_Y)]

# optimal cut-off that minimizes the NSR criterion
results <- EWS_NSR_Criterion(Var_Proba = vector_proba, Dicho_Y = vector_binary,
                              cutoff_interval = 0.0001)

# print results
results

#}

---

GIRF_Dicho

**GIRF for Dichotomous models**

**Description**

This function estimates the response functions of dichotomous models in a univariate framework using the method proposed by Lajaunie (2021). The response functions are based on the 4 specifications proposed by Kauppi & Saikkonen (2008).

**Usage**

GIRF_Dicho(Dicho_Y, Exp_X, Lag, Int, t_mod, horizon, shock_size, OC)

**Arguments**

- **Dicho_Y** Vector of the binary time series.
- **Exp_X** Vector or Matrix of explanatory time series.
- **Lag** Number of lags used for the estimation.
- **Int** Boolean value: TRUE for an estimation with intercept, and FALSE otherwise.
- **t_mod** Model number: 1, 2, 3 or 4. 
  -> 1 for the static model:
\[ P_{t-1}(Y_t) = F(\pi_t) = F(\alpha + \beta' X_t) \]

-> 2 for the dynamic model with lag binary variable:

\[ P_{t-1}(Y_t) = F(\pi_t) = F(\alpha + \beta' X_t + \gamma Y_{t-1}) \]

-> 3 for the dynamic model with lag index variable:

\[ P_{t-1}(Y_t) = F(\pi_t) = F(\alpha + \beta' X_t + \eta \pi_{t-1}) \]

-> 4 for the dynamic model with both lag binary variable and lag index variable:

\[ P_{t-1}(Y_t) = F(\pi_t) = F(\alpha + \beta' X_t + \eta \pi_{t-1} + \gamma Y_{t-1}) \]

**horizon**
Numeric variable corresponding to the horizon target for the GIRF analysis.

**shock_size**
Numeric variable equal to the size of the shock. It can be estimated with the Vector_Error function.

**OC**
Numeric variable equal to the Optimal Cut-off (threshold). This threshold can be considered arbitrarily, with a value between 0 and 1, or it can be estimated with one of the functions EWS_AM_Criterion, EWS-CSA_Criterion, or EWS_NSR_Criterion.

**Value**
Matrix with 7 columns:

- column 1: horizon
- column 2: index
- column 3: index with shock
- column 4: probability associated to the index
- column 5: probability associated to the index with shock
- column 6: binary variable associated to the index
- column 7: binary variable associated to the index with shock

**Author(s)**
Jean-Baptiste Hasse and Quentin Lajaunie

**References**

Examples

```r
# NOT RUN {
# Import data
data("data_USA")

# Data process
Var_Y <- as.vector(data_USA$NBER)
Var_X <- as.vector(data_USA$Spread)

# Estimate the logit regression
Logistic_results <- Logistic_Estimation(Dicho_Y = Var_Y, Exp_X = Var_X, Intercept = TRUE,
                                        Nb_Id = 1, Lag = 1, type_model = 1)

# Vector of probabilities
vector_proba <- as.vector(rep(0,length(Var_Y)-1))
vector_proba <- Logistic_results$prob

# Vector of binary variables
Lag <- 1
vector_binary <- as.vector(rep(0,length(Var_Y)-1))
vector_binary <- Var_Y[(1+Lag):length(Var_Y)]

# optimal cut-off that maximizes the AM criterion
Threshold_AM <- EWS_AM_Criterion(Var_Proba = vector_proba, Dicho_Y = vector_binary,
                                  cutoff_interval = 0.0001)

# Estimate the estimation errors
Residuals <- Vector_Error(Dicho_Y = Var_Y, Exp_X = Var_X, Intercept = TRUE,
                           Nb_Id = 1, Lag = 1, type_model = 1)

# Initialize the shock
size_shock <- quantile(Residuals,0.95)

# GIRF Analysis
results <- GIRF_Dicho(Dicho_Y = Var_Y, Exp_X = Var_X, Lag = 1, Int = TRUE, t_mod = 1,
                      horizon = 3, shock_size = size_shock, OC = Threshold_AM)

# print results
results
#
```

Description

From the results of the Simulation_GIRF function, this function calculates the values of the upper and lower bounds of the confidence intervals, as well as the average of the different response
functions for the index.

Usage

GIRF_Index_CI(results_simul_GIRF, CI_bounds, n_simul, horizon_forecast)

Arguments

results_simul_GIRF
Matrix containing results of the Simulation_GIRF function.

CI_bounds
Numeric variable between 0 and 1 for the size of the confidence intervals.

n_simul
Numeric variable equal to the total number of replications.

horizon_forecast
Numeric variable corresponding to the horizon target for the GIRF analysis.

Value

A list with:

Simulation_CI
a matrix containing the set of simulations belonging to the confidence interval.

values_CI
a matrix containing three columns: the lower bound of the CI, the average of the IRFs, and the upper bound of the CI.

Author(s)

Jean-Baptiste Hasse and Quentin Lajaunie

References


Examples

# NOT RUN {

# Import data
data("data_USA")

# Data process
Var_Y <- as.vector(data_USA$NBER)
Var_X <- as.vector(data_USA$Spread)

# Simulation for the GIRF analysis
results_simulation <- Simul_GIRF(Var_Y, Var_X, TRUE, 1, 1, 2, 0.95, 3, "AM")

# Confidence intervals for the index
results <- GIRF_Index_CI(results_simulation, 0.95, 2, 3)
# print results
results
#

GIRF_Proba_CI  Confidence Intervals for the Probability - GIRF Analysis

Description

From the results of the Simulation_GIRF function, this function calculates the values of the upper and lower bounds of the confidence intervals, as well as the average of the different response functions for the probability.

Usage

GIRF_Proba_CI(results_simul_GIRF, CI_bounds, n_simul, horizon_forecast)

Arguments

results_simul_GIRF
   Matrix containing results of the Simulation_GIRF function.
CI_bounds
   Numeric variable between 0 and 1 for the size of the confidence intervals.
n_simul
   Numeric variable equal to the total number of replications.
horizon_forecast
   Numeric variable corresponding to the horizon target for the GIRF analysis.

Value

A list with:

horizon
   Numeric vector containing the horizon target.
Simulation_CI_proba_shock
   a matrix containing the set of simulations of probabilities with shock belonging to the confidence interval.
Simulation_CI_proba
   a matrix containing the set of simulations of probabilities belonging to the confidence interval.
CI_proba_shock
   a matrix containing three columns: the lower bound of the CI, the average of the IRFs, and the upper bound of the CI for the probabilities with shock.
CI_proba
   a matrix containing three columns: the lower bound of the CI, the average of the IRFs, and the upper bound of the CI for the probabilities.

Author(s)

Jean-Baptiste Hasse and Quentin Lajaunie
Logistic_Estimation

References


Examples

```r
# NOT RUN {

# Import data
data("data_USA")

# Data process
Var_Y <- as.vector(data_USA$NBER)
Var_X <- as.vector(data_USA$Spread)

# Simulation for the GIRF analysis
results_simulation <- Simul_GIRF(Var_Y, Var_X, TRUE, 1, 1, 2, 0.95, 3, "AM")

# Confidence intervals for the index
results <- GIRF_Proba_CI(results_simulation, 0.95, 2, 3)

# print results
results

#}
```

Logistic_Estimation  

Logistic Estimation for Dichotomous Analysis

Description

This function provides methods for estimating the four dichotomous models as in Kauppi & Saikkonen (2008). Based on a logit approach, models are estimated in a univariate or a balanced panel framework as in Candelon, Dumitrescu and Hurlin (2014). This estimation has been used in recent papers such in Ben Naceur, Candelon and Lajaunie (2019) and Hasse and Lajaunie (2020).

Usage

`Logistic_Estimation(Dicho_Y, Exp_X, Intercept, Nb_Id, Lag, type_model)`

Arguments

- `Dicho_Y`  
  Vector of the binary time series.
- `Exp_X`  
  Vector or Matrix of explanatory time series.
- `Intercept`  
  Boolean value: TRUE for an estimation with intercept, and FALSE otherwise.
Number of individuals studied for a panel approach. \( \text{Nb}_{-}\text{Id}=1 \) in the univariate case.

Number of lags used for the estimation.

Model number: 1, 2, 3 or 4.

\[ P_{t-1}(Y_t) = F(\pi_t) = F(\alpha + \beta'X_t) \]

\[ P_{t-1}(Y_t) = F(\pi_t) = F(\alpha + \beta'X_t + \gamma Y_{t-1}) \]

\[ P_{t-1}(Y_t) = F(\pi_t) = F(\alpha + \beta'X_t + \eta \pi_{t-1}) \]

\[ P_{t-1}(Y_t) = F(\pi_t) = F(\alpha + \beta'X_t + \eta \pi_{t-1} + \gamma Y_{t-1}) \]

A list with:

- **Estimation**: a dataframe containing the coefficients of the logistic estimation, the Standard Error for each coefficient, the Z-score and the associated critical probability
- **AIC**: a numeric vector containing the Akaike information criterion
- **BIC**: a numeric vector containing the Bayesian information criterion
- **R2**: a numeric vector containing the Pseudo R Square
- **index**: a numeric vector containing the estimated index
- **prob**: a numeric vector containing the estimated probabilities
- **LogLik**: a numeric vector containing the Log likelihood value of the estimation
- **VCM**: a numeric matrix of the Variance Covariance of the estimation

For the panel estimation, data must be stacked one after the other for each country or for each individual.

**Author(s)**

Jean-Baptiste Hasse and Quentin Lajaunie
References


Examples

```r
# NOT RUN {
# Import data
data("data_USA")

# Data process
Var_Y <- as.vector(data_USA$NBER)
Var_X <- as.vector(data_USA$Spread)

# Estimate the logit regression
results <- Logistic_Estimation(Dicho_Y = Var_Y, Exp_X = Var_X, Intercept = TRUE,
                               Nb_Id = 1, Lag = 1, type_model = 1)

# print results
results

# }

Matrix_lag

Matrix Lag - data processing

Description

Compute a lagged version of a time series, shifting the time base back by a given number of observations defined by the user. The user must enter three parameters for this function: the matrix, the number of lags, and of boolean variable calls 'beginning'. If 'beginning'=TRUE, then the lag will be applied at the beginning of the matrix whereas if 'beginning'=FALSE, then the lag will be applied at the end of the matrix.

Usage

Matrix_lag(Matrix_target, Nb_lag, beginning)
Simul_GIRF

Arguments

Matrix_target  Initial Matrix
Nb_lag          Number of lag
beginning       Boolean variable. If 'place'=TRUE, the lag is applied at the beginning of the matrix. If 'place'=FALSE, the lag is applied at the end of the matrix.

Value

A numeric Matrix.

Examples

# NOT RUN {

# Initialize the following matrix
Matrix_example <- matrix(data=(1:10), nrow=5, ncol=2)

# Use Matrix_lag
new_matrix <- Matrix_lag(Matrix_target = Matrix_example, Nb_lag = 1, beginning = TRUE)

new_matrix

# Results:
#> new_matrix
#>      [,1] [,2]
#> [1,]   2   7
#> [2,]   3   8
#> [3,]   4   9
#> [4,]   5  10

#}

Simul_GIRF  GIRF Simulations

Description

This function calls the BlockBootstrap function of the EWS package and then calculates response functions for each simulation. It then measures the confidence intervals as in Lajaunie (2021). The response functions are based on the 4 specifications proposed by Kauppi & Saikkonen (2008).

Usage

Simul_GIRF(Dicho_Y, Exp_X, Int, Lag, t_mod, n_simul, centile_shock, horizon, OC)
Arguments

- **Dicho_Y**: Vector of the binary time series.
- **Exp_X**: Vector or Matrix of explanatory time series.
- **Int**: Boolean value: TRUE for an estimation with intercept, and FALSE otherwise.
- **Lag**: Number of lags used for the estimation.
- **t_mod**: Model number: 1, 2, 3 or 4.
  - -> 1 for the static model:
    \[ P_{t-1}(Y_t) = F(\pi_t) = F(\alpha + \beta'X_t) \]
  - -> 2 for the dynamic model with lag binary variable:
    \[ P_{t-1}(Y_t) = F(\pi_t) = F(\alpha + \beta'X_t + \gamma Y_{t-l}) \]
  - -> 3 for the dynamic model with lag index variable:
    \[ P_{t-1}(Y_t) = F(\pi_t) = F(\alpha + \beta'X_t + \eta \pi_{t-l}) \]
  - -> 4 for the dynamic model with both lag binary variable and lag index variable:
    \[ P_{t-1}(Y_t) = F(\pi_t) = F(\alpha + \beta'X_t + \eta \pi_{t-l} + \gamma Y_{t-l}) \]
- **n_simul**: Numeric variable equal to the total number of replications.
- **centile_shock**: Numeric variable corresponding to the centile of the shock following Koop, Pesaran and Potter (1996).
- **horizon**: Numeric variable corresponding to the horizon target for the GIRF analysis.
- **OC**: Either a numeric variable equal to the optimal cut-off (threshold) or a character variable of the method chosen to calculate the optimal cut-off ("NSR", "CSA", "AM").

Value

A matrix containing the GIRF analysis for each replication. For each replication, the function returns 7 columns with:

- column 1: horizon
- column 2: index
- column 3: index with shock
- column 4: probability associated to the index
- column 5: probability associated to the index with shock
- column 6: binary variable associated to the index
- column 7: binary variable associated to the index with shock

The matrix contains \(7 \times S\) columns, where \(S\) denotes the number of replications.
Author(s)

Jean-Baptiste Hasse and Quentin Lajaunie

References


Examples

# NOT RUN {

# Import data
data("data_USA")

# Data process
Var_Y <- as.vector(data_USA$NBER)
Var_X <- as.vector(data_USA$Spread)

# Simulations
results <- Simul_GIRF(Dicho_Y = Var_Y, Exp_X = Var_X, Int = TRUE, Lag = 1, t_mod = 1, n_simul = 2, centile_shock = 0.95, horizon = 3, OC = "AM")

# print results
results

#}

---

Vector_Error  Vector of Errors

Description

The function measures the estimation errors from the logistic estimation, and stores them in a vector. This function is used to initialize a shock in impulse response analysis as in Koop, Pesaran and Potter (1996).

Usage

Vector_Error(Dicho_Y, Exp_X, Intercept, Nb_Id, Lag, type_model)
Vector_Error

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dicho_Y</td>
<td>Vector of the binary time series.</td>
</tr>
<tr>
<td>Exp_X</td>
<td>Vector or Matrix of explanatory time series.</td>
</tr>
<tr>
<td>Intercept</td>
<td>Boolean value: TRUE for an estimation with intercept, and FALSE otherwise.</td>
</tr>
<tr>
<td>Nb_Id</td>
<td>Number of individuals studied for a panel approach. Nb_Id=1 in the univariate case.</td>
</tr>
<tr>
<td>Lag</td>
<td>Number of lags used for the estimation.</td>
</tr>
<tr>
<td>type_model</td>
<td>Model number: 1, 2, 3 or 4.</td>
</tr>
</tbody>
</table>

**Value**

A numeric vector containing estimation errors.

**Author(s)**

Jean-Baptiste Hasse and Quentin Lajaunie

**References**


**Examples**

```r
# NOT RUN {

# Import data
data("data_USA")

# Data process
Var_Y <- as.vector(data_USA$NBER)
Var_X <- as.vector(data_USA$Spread)

# Estimate the estimation errors
results <- Vector_Error(Dicho_Y = Var_Y, Exp_X = Var_X, Intercept = TRUE,
                         Nb_Id = 1, Lag = 1, type_model = 4)

# print results
results

#}
```
Vector_lag

Vector_lag - data processing

Description

Compute a lagged version of a time series, shifting the time base back by a given number of observations defined by the user. The user must enter three parameters for this function: the vector, the number of lags, and a boolean variable named 'beginning'. If 'beginning'=TRUE, then the lag will be applied at the beginning of the vector whereas if 'beginning'=FALSE, then the lag will be applied at the end of the vector.

Usage

Vector_lag(Vector_target, Nb_lag, beginning)

Arguments

- Vector_target: Initial vector
- Nb_lag: Number of lag
- beginning: Boolean variable. If 'beginning'=TRUE, the lag is applied at the beginning of the vector. If 'beginning'=FALSE, the lag is applied at the end of the vector.

Value

A numeric Vector.

Examples

# NOT RUN {

# Initialize the following vector
vector_example <- as.vector(1:10)

# Use Vector_lag
new_vector <- Vector_lag(Vector_target = vector_example, Nb_lag = 2, beginning = TRUE)

new_vector
# Results:
#> new_vector
#> [1] 3 4 5 6 7 8 9 10

#}

Index

* **Bootstrapp**
  BlockBootstrapp, 2
* **Confidence-Intervals**
  BlockBootstrapp, 2
  GIRF_Index_CI, 11
  GIRF_Proba_CI, 13
  Simul_GIRF, 17
* **Dichotomous**
  EWS_AM_Criterion, 5
  EWS_CSA_Criterion, 6
  EWS_NSR_Criterion, 8
  GIRF_Dicho, 9
  GIRF_Index_CI, 11
  GIRF_Proba_CI, 13
  Logistic_Estimation, 14
  Simul_GIRF, 17
  Vector_Error, 19
* **Econometrics**
  BlockBootstrapp, 2
  EWS_AM_Criterion, 5
  EWS_CSA_Criterion, 6
  EWS_NSR_Criterion, 8
  GIRF_Dicho, 9
  GIRF_Index_CI, 11
  GIRF_Proba_CI, 13
  Logistic_Estimation, 14
  Simul_GIRF, 17
  Vector_Error, 19
* **IRF**
  GIRF_Dicho, 9
  GIRF_Index_CI, 11
  GIRF_Proba_CI, 13
  Simul_GIRF, 17
* **Panel**
  Logistic_Estimation, 14
* **Shock**
  Vector_Error, 19
* **Threshold**
  EWS_AM_Criterion, 5

* **Datasets**
  data_panel, 3
  data_USA, 4
  BlockBootstrapp, 2
  data_panel, 3
  data_USA, 4
  EWS_AM_Criterion, 5
  EWS_CSA_Criterion, 6
  EWS_NSR_Criterion, 8
  GIRF_Dicho, 9
  GIRF_Index_CI, 11
  GIRF_Proba_CI, 13
  Logistic_Estimation, 14
  Matrix_lag, 16
  Simul_GIRF, 17
  Vector_Error, 19
  Vector_lag, 21