Package ‘MultiATSM’

November 23, 2023

Type Package
Title Multicountry Term Structure of Interest Rates Models
Version 0.3.5
Date 2023-11-23
Author Rubens Moura
Maintainer Rubens Moura <rubens.gtmoura@gmail.com>
License GPL-2 | GPL-3
Encoding UTF-8
RoxygenNote 7.1.1
Imports zoo, pracma, wrapr, hablar, ggplot2
Suggests readxl, readr, magic, Jmisc, functional, cowplot, reshape2,
    sjmisc, stringr, knitr, rmarkdown, bookdown, kableExtra,
    neldermead, magrittr
Depends R (>= 3.5.0)
VignetteBuilder knitr
NeedsCompilation no
Repository CRAN
Date/Publication 2023-11-23 07:10:02 UTC

R topics documented:

  Bias_Correc_VAR ......................................................... 2
  Bootstrap ............................................................... 5
Bias_Correc_VAR

Estimate an unbiased VAR(1) using stochastic approximation (Bauer, Rudebusch and Wu, 2012)

Index

Bias_Correc_VAR    Estimate an unbiased VAR(1) using stochastic approximation (Bauer, Rudebusch and Wu, 2012)

Description

Estimate an unbiased VAR(1) using stochastic approximation (Bauer, Rudebusch and Wu, 2012)
Bias_Correc_VAR

Usage

Bias_Correc_VAR(
    ModelType,  
    BRWinputs,  
    RiskFactors,  
    N,  
    Economies,  
    FactorLabels,  
    GVARinputs = NULL,  
    JLLinputs = NULL,  
    ev_restr = 1,  
    nargout = 4
)

Arguments

ModelType          string-vector containing the label of the model to be estimated
BRWinputs          List containing the following necessary inputs for the estimation of the BRW model:
                     1. flag_mean: flag whether mean- (TRUE) or median- (FALSE) unbiased estimation is desired. Default is set to TRUE;
                     2. gamma: adjustment parameter. Value parameters should vary between 0 and 1. Default is set to 0.5;
                     3. N_iter: number of iterations used in the stochastic approximation algorithm after burn-in. Default is set to 5,000;
                     4. N_burn: number of burn-in iterations used in the stochastic approximation algorithm. Default is set to 0.15*N_iter;
                     5. B: number of bootstrap samples per iteration to calculate noisy measure of mean/median of the OLS estimator. Default is set to 50;
                     6. check: flag whether the user wishes to perform the closeness check. Default is set to TRUE;
                     7. B_check: number of bootstrap samples used in the closeness check. Default is set to 100,000.
RiskFactors        time series of the risk factors (T x F)
N                   number of country-specific spanned factors (scalar)
Economies          string-vector containing the names of the economies which are part of the economic system
FactorLabels       string-list based which contains the labels of all variables present in the model
GVARinputs         inputs used in the estimation of the GVAR-based models (see "GVAR" function). Default is set to NULL
JLLinputs          inputs used in the estimation of the JLL-based models (see "JLL" function). Default is set to NULL
ev_restr           largest eigenvalue restriction under the P-measure. Default is set to 1
nargout            number of elements present in the list of outputs. Default is set to 4
Bias-corrected VAR parameters based on the framework of Bauer, Rudebusch and Wu (2012). The list contains:

1. \( \Phi_{\text{tilde}} \): estimated coefficient matrix (\( F \times F \));
2. \( \mu_{\text{tilde}} \): estimated intercept (\( F \times 1 \));
3. \( V_{\text{tilde}} \): estimated variance-covariance matrix (\( F \times F \));
4. \( \text{dist} \): root mean square distance (scalar);
5. \( \Phi_{\text{sample}} \): sample estimated variance-covariance matrix used in the checks (\( F \times F \times B_{\text{check}} \)) - this output is reported if \( nargout \) is set to 5.

References

Bauer, Rudebusch and, Wu (2012). "Correcting Estimation Bias in Dynamic Term Structure Models"  
This function is based on the "est_unb_var" Matlab function available at Cynthia Wu’s website (https://sites.google.com/view/jingcynthiawu/).

Examples

data(CM_Factors)
Factors <- t(RiskFactors[1:7,])

BRWinputs <- list()
BRWinputs$flag_mean <- TRUE
BRWinputs$gamma <- 0.4
BRWinputs$N_iter <- 1000
BRWinputs$N_burn <- 100
BRWinputs$B <- 10
BRWinputs$check <- 1
BRWinputs$B_check <- 5000

Economies <- "China"
N <- 3
ModelType <- "JPS"
FactorLabels <- NULL

BRWpara <- Bias_Correc_VAR(ModelType, BRWinputs, Factors, N, Economies, FactorLabels)
Bootstrap  

Generates the bootstrap-related outputs

Description

Generates the bootstrap-related outputs

Usage

Bootstrap(
    ModelType,
    ModelParaPE,
    NumOutPE,
    mat,
    Economies,
    InputsForOutputs,
    FactorLabels,
    DataFrequency,
    vararginPE,
    JLLinputs = NULL,
    GVARinputs = NULL,
    BRWinputs = NULL
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModelType</td>
<td>string-vector containing the label of the model to be estimated</td>
</tr>
<tr>
<td>ModelParaPE</td>
<td>point estimate from the model parameters (see the outputs of the &quot;Optimization&quot; function)</td>
</tr>
<tr>
<td>NumOutPE</td>
<td>point estimate from the numerical outputs (see the outputs of the &quot;NumOutputs&quot; function)</td>
</tr>
<tr>
<td>mat</td>
<td>vector of maturities (in years) used in the estimation</td>
</tr>
<tr>
<td>Economies</td>
<td>string-vector containing the names of the economies which are part of the economic system</td>
</tr>
<tr>
<td>InputsForOutputs</td>
<td>list containing the desired inputs for the construction of IRFs, GIRFs, FEVDs, and GFEVDs.</td>
</tr>
<tr>
<td>FactorLabels</td>
<td>string-list based which contains the labels of all the variables present in the model</td>
</tr>
<tr>
<td>vararginPE</td>
<td>list containing starting values and constraints (see arguments of the &quot;Optimization&quot; function)</td>
</tr>
<tr>
<td>JLLinputs</td>
<td>list of necessary inputs for the estimation of JLL-based models (see &quot;JLL&quot; function)</td>
</tr>
</tbody>
</table>
GVARinputs  list of necessary inputs for the estimation of GVAR-based models (see "GVAR" function)
BRWinputs  list of necessary inputs for performing the bias-corrected estimation (see "Bias_Correc_VAR" function)

Value
list containing the following elements:

- list of model parameters for one each one the draws;
- list of numerical outputs (IRFs, GIRFs, FEVDs, GFEVDs) for each one of the draws;
- Confidence bands for the chosen level of significance.

References
This function is a modified and extended version of the "VARirbound" function from "A toolbox for VAR analysis" by Ambrogio Cesa-Bianchi (https://github.com/ambropo/VAR-Toolbox)

Examples
# See examples in the vignette file of this package (Section 4).

---

BR_jps_out  Replications of the JPS (2014) outputs by Bauer and Rudebusch (2017)

Description
Unspanned macro risk model outputs by Bauer and Rudebusch (2017)

Usage
data("BR_jps_gro_R3")

Format
Unspanned macro risk model outputs by Bauer and Rudebusch (2017)
est.llk  summary list of log-likelihood estimations
M.o  time series of unspanned factors
pars  additional summary list of log-likelihood estimations
W  Weight matrix that results from principal components analysis
Y  time series of bond yields
N  total number of risk factor of the model (spanned and unspanned)
R  total number of spanned factor of the model
DatabasePrep

References

Bauer, M. and Rudebusch, G. "Resolving the Spanning Puzzle in Macro-Finance Term Structure Models"

DatabasePrep Prepare the GVARFactors database

Description

Prepare the GVARFactors database

Usage

DatabasePrep(
  t_First,  
  t_Last,  
  Economies,  
  N,  
  FactorLabels,  
  ModelType,  
  Wgvar = NULL,  
  DataPathMacro = NULL,  
  DataPathYields = NULL  
)

Arguments

  t_First  sample starting date (yyyy-mm-dd)  
  t_Last  sample last date (yyyy-mm-dd)  
  Economies  string-vector containing the names of the economies which are part of the economic system  
  N  number of country-specific spanned factor (scalar)  
  FactorLabels  list containing the factor labels  
  ModelType  string-vector containing the label of the model to be estimated  
  Wgvar  GVAR transition matrix (CxC), if GVAR type model is chosen; default is set to NULL.  
  DataPathMacro  path of the Excel file containing the macroeconomic data (if any). The default is linked to the Excel file available in the package.  
  DataPathYields  path of the Excel file containing the yields data (if any). The default is linked to the Excel file available in the package.

Value

List of the risk factor set used in the estimation of the GVAR model  
List containing the risk factor set used in the estimation of the GVAR-based models
Examples

```r
DomVar <- c("Eco_Act", "Inflation")
GlobalVar <- c("GBC", "CPI_OECD")
t0 <- "2006-09-01"
tF <- "2019-01-01"
Economies <- c("China", "Brazil", "Mexico", "Uruguay", "Russia")
N <- 3
ModelType <- "JPS jointQ"
FactorLabels <- LabFac(N, DomVar, GlobalVar, Economies, ModelType)

GVARFactors <- DatabasePrep(t0, tF, Economies, N, FactorLabels, ModelType)
```

---

**DataForEstimation**

*Retrieve data from Excel and build the database used in the model estimation*

**Description**

Retrieve data from Excel and build the database used in the model estimation

**Usage**

```r
DataForEstimation( 
  t0, 
  tF, 
  Economies, 
  N, 
  FactorLabels, 
  ModelType, 
  DataFrequency, 
  W_type = NULL, 
  t_First_Wgvar = NULL, 
  t_Last_Wgvar = NULL, 
  DataPathMacro = NULL, 
  DataPathYields = NULL, 
  DataPathTrade = NULL 
)
```

**Arguments**

- **t0**  
  Sample starting date (yyyy-mm-dd)
- **tF**  
  Sample last date (yyyy-mm-dd)
- **Economies**  
  string-vector containing the names of the economies which are part of the economic system
- **N**  
  Number of country-specific spanned factor (scalar)
DataForEstimation

FactorLabels String-list based which contains the labels of all the variables present in the model

ModelType String-vector containing the label of the model to be estimated


W_type Three possibilities:
  • "Full Sample": if one wishes ALL weight matrices of each year from which data is available (it may extrapolate the sample period);
  • "Sample Mean": if one wishes a SINGLE weight matrix containing the average of weights over of the entire sample period;
  • Some year in particular (e.g. "1998", "2005" ...).

t_First_Wgvar Sample starting date (year)

t_Last_Wgvar Sample last date (year)

DataPathMacro Path of the Excel file containing the macroeconomic data (if any). The default is linked to the excel file present in the package.

DataPathYields Path of the Excel file containing the yields data (if any). The default is linked to the excel file present in the package.

DataPathTrade Path of the Excel file containing the trade data (if any). The default is linked to the excel file present in the package.

Value

A list containing the

1. time series of the complete set of bond yields (matrix, JxT or CJxT);
2. time series of the complete set risk factors (matrix, KxT);
3. 'GVARFactors': list of all variables that are used in the estimation of the VARX (see e.g. 'CM_Factors_GVAR' file). If the estimated model type is not GVAR-based, then returns NULL.

Examples

DomVar <- c("Eco_Act", "Inflation")
GlobalVar <- c("GBC", "CPI_OECD")
t0 <- "2006-09-01"
tF <- "2019-01-01"
Economies <- c("China", "Brazil", "Mexico", "Uruguay", "Russia")
N <- 2
ModelType <- "JPS"
FactorLabels <- LabFac(N, DomVar, GlobalVar, Economies, ModelType)
DataFrequency <- "Monthly"

DataModel <- DataForEstimation(t0, tF, Economies, N, FactorLabels, ModelType, DataFrequency)
FactorsGVAR

**Data:** Risk Factors for the GVAR - Candelon and Moura (2021)

**Description**
Risk factors data used in the GVAR models - Candelon and Moura (2021)

**Usage**
```
data("CM_Factors_GVAR")
```

**Format**
list containing the variables used in the GVAR models

**References**
Candelon, B. and Moura, R. "A Multi-Country Model of the Term Structures of Interest Rates with a GVAR".

---

ForecastYields

**Gather bond yields forecasts for all the model types**

**Description**
Gather bond yields forecasts for all the model types

**Usage**
```
ForecastYields(
    ModelType,
    ModelPara,
    InputsForOutputs,
    FactorLabels,
    Economies,
    DataFrequency,
    JLLinputs,
    GVARinputs,
    BRWinputs
)
```
Function

**Arguments**

- **ModelType**
  - a string-vector containing the label of the model to be estimated
- **ModelPara**
  - List of model parameter estimates (See the "Optimization" function)
- **InputsForOutputs**
  - list containing the desired horizon of analysis for the IRFs, GIRFs, FEVDs, and GFEVDs
- **FactorLabels**
  - a string-list based which contains all the labels of all the variables present in the model
- **Economies**
  - string-vector containing the names of the economies which are part of the economic system
- **DataFrequency**
- **JLLinputs**
  - list of necessary inputs for the estimation of JLL-based models (see "JLL" function)
- **GVARinputs**
  - list of necessary inputs for the estimation of GVAR-based models (see "GVAR" function)
- **BRWinputs**
  - list of necessary inputs for performing the bias-corrected estimation (see "Bias_Correc_VAR" function)

**Value**

List containing the following elements

1. Out-of-sample forecasts of bond yields per forecast horizon
2. Out-of-sample forecast errors of bond yields per forecast horizon
3. Root mean square errors per forecast horizon

**Examples**

# See examples in the vignette file of this package (Section 4).

| Functionf | Set up the vector-valued objective function (Point estimate) |

**Description**

Set up the vector-valued objective function (Point estimate)

**Usage**

Functionf(MLEinputs, Economies, mat, DataFrequency, FactorLabels, ModelType)
Arguments

MLEinputs  Set of inputs that are necessary to the log-likelihood function
Economies  string-vector containing the names of the economies which are part of the e-
            conomic system
mat        vector of maturities (in years) of yields used in estimation (J x 1)
                "Monthly", "Quarterly", "Annually"
FactorLabels string-list based which contains the labels of all the variables present in the
               model
ModelType   string-vector containing the label of the model to be estimated

Value

objective function

Examples

# See examples in the vignette file of this package (Section 4).

GVAR

Estimate a GVAR(1) and a VARX(1,1,1)

Description

Estimate a GVAR(1) and a VARX(1,1,1)

Usage

GVAR(GVARinputs, N)

Arguments

GVARinputs  List containing the following necessary inputs for the estimation of the GVAR:
            1. Economies: string-vector containing the names of the economies which are
               part of the economic system
            2. 'GVARFactors': list of all variables that are used in the estimation of the
               VARX
               (see e.g. 'CM_Factors_GVAR' file);
            3. 'VARXtype': character-vector containing three possibilities:
               • 'unconstrained': model is estimated without any constrained (each equa-
               tion is estimated individually by OLS);
               • 'constrained: Spanned Factors': model is estimated taking into account
               the fact that foreign-pricing-factors do NOT impinge on (i) domestic
               economic variables and (ii) domestic pricing factors. (equations are
               estimated by restricted least squares)
• ‘constrained: ’ extended by the name of the risk factor: model is estimated taking into account the fact that the restricted factor is only affected by its own lagged values and the lagged values of its own star variables. (equations are estimated by restricted least squares)

4. 'Wgvar': GVAR transition matrix (C x C) - see the output from 'Transition_Matrix' function

N number of country-specific spanned factors (scalar)

Value

A list containing

1. parameters of the country-specific VARX(1,1,1)
   • intercept (M+Nx1);
   • phi_1 (M+N x M+N);
   • phi_1^star (M+N x M+N);
   • phi_g (M+N x M+N);
   • Sigma (M+N x G)

2. parameters of the GVAR.
   • F0 (F X 1);
   • F1 (F x F);
   • Sigma_y (F x F)

References


Examples

data(CM_Factors_GVAR)

N <- 3

GVARinputs <- list()
GVARinputs$Economies <- c("China", "Brazil", "Mexico", "Uruguay")
GVARinputs$GVARFactors <- FactorsGVAR
GVARinputs$VARXtype <- "unconstrained"
GVARinputs$Wgvar <- matrix( c(0, 0.83, 0.86, 0.38, 
                               0.65, 0, 0.13, 0.55, 
                               0.32, 0.12, 0, 0.07, 
                               0.03, 0.05, 0.01, 0), nrow = 4, ncol = 4)

GVAR(GVARinputs, N)
**GVARFactors**  
*Data: Risk Factors for the GVAR - Candelon and Moura (2023)*

**Description**
Risk factors data used in the GVAR models - Candelon and Moura (2023)

**Usage**
```r
data("CM_Factors_GVAR_2023")
```

**Format**
list containing the variables used in the GVAR models

**References**
Candelon, B. and Moura, R. "Sovereign yield curves and the COVID-19 in emerging markets".

---

**InputsForMLEdensity**  
*Generates several inputs that are necessary to build the likelihood function*

**Description**
Generates several inputs that are necessary to build the likelihood function

**Usage**
```r
InputsForMLEdensity(
    ModelType, 
    Yields, 
    PdynamicsFactors, 
    FactorLabels, 
    mat, 
    Economies, 
    DataFrequency, 
    JLLinputs = NULL, 
    GVARinputs = NULL, 
    BRWinputs = NULL
)
```
## Arguments

**ModelType**
- string-vector containing the label of the model to be estimated

**Yields**
- time series of yields (JxT or CJ x T)

**PdynamicsFactors**
- time series of the risk factors (K x T)

**FactorLabels**
- string-list based which contains the labels of all variables present in the model

**mat**
- vector of maturities (in years) used in the estimation

**Economies**
- string-vector containing the names of the economies of the system.
  - If the ModelType selected is "JPS", "JPS jointP", "GVAR sepQ", then only one economy can be selected.
  - For the other models, more than one economy must be selected.

**DataFrequency**

**JLLinputs**
- list of necessary inputs for the estimation of JLL-based models (see "JLL" function)

**GVARinputs**
- list of necessary inputs for the estimation of GVAR-based models (see "GVAR" function)

**BRWinputs**
- list of necessary inputs for performing the bias-corrected estimation (see "Bias_Correc_VAR" function)

## Details

To ensure that the risk factors matrix is correctly built for the model "JPS", the global factors should be allocated on the first G rows of this matrix.

## Value

List of necessary inputs for constructing the model’s log-likelihood function

## Examples

```r
# Example 1:
data(CM_Factors)
data(CM_Yields)

ModelType <- "JPS"
Economies <- "Mexico"
Factors <- RiskFactors
N <- 3
GlobalVar <- c("GBC", "CPI_OECD") # Global Variables
DomVar <- c("Eco_Act", "Inflation") # Domestic Variables
FactorLabels <- LabFac(N, DomVar, GlobalVar, Economies, ModelType)

mat <- c(0.25, 0.5, 1, 3, 5, 10)
DataFrequency <- "Monthly"
```
i <- length(Economies)
ATSMInputs <- InputsForMLEdensity(ModelType, Yields, Factors, FactorLabels, mat, Economies, DataFrequency)

# Example 2:
data(CM_Factors)
data(CM_Yields)
data(CM_Factors_GVAR)

ModelType <- "GVAR jointQ"
Economies <- c("China", "Brazil", "Mexico", "Uruguay")
mat <- c(0.25, 0.5, 1, 3, 5, 10)
DataFrequency <- "Monthly"
Factors <- RiskFactors
N <- 3
GlobalVar <- c("GBC", "CPI_OECD") # Global Variables
DomVar <- c("Eco_Act", "Inflation") # Domestic Variables
FactorLabels <- LabFac(N, DomVar, GlobalVar, Economies, ModelType)

GVARinputs <- list()
GVARinputs$Economies <- Economies
GVARinputs$GVARFactors <- FactorsGVAR
GVARinputs$VARXtype <- "unconstrained"
GVARinputs$Wgvar <- matrix(c(0, 0.83, 0.86, 0.38,
                             0.65, 0, 0.13, 0.55,
                             0.32, 0.12, 0, 0.07,
                             0.03, 0.05, 0.01, 0), nrow = 4, ncol = 4)
ATSMInputs <- InputsForMLEdensity(ModelType, Yields, Factors, FactorLabels, mat, Economies, DataFrequency, JLLinputs = NULL, GVARinputs)

# Example 3:
if (requireNamespace("neldermead", quietly = TRUE)) {

data(CM_Factors)
data(CM_Yields)
ModelType <- "JLL jointSigma"
GlobalVar <- c("GBC", "CPI_OECD") # Global Variables
DomVar <- c("Eco_Act", "Inflation") # Domestic Variables
N <- 3
Economies <- c("China", "Brazil", "Mexico", "Uruguay")
FactorLabels <- LabFac(N, DomVar, GlobalVar, Economies, ModelType)

Factors <- RiskFactors
mat <- c(0.25, 0.5, 1, 3, 5, 10)
DataFrequency <- "Monthly"
JLLinputs <- list()
JLLinputs$Economies <- Economies
JLLinputs$DomUnit <- "China"
JLLinputs$WishSigmas <- 1
JLLinputs$SigmaNonOrtho <- NULL
JLLinputs$JLLModelType <- ModelType

ATSMInputs <- InputsForMLEdensity(ModelType, Yields, Factors, FactorLabels, mat, Economies, DataFrequency, JLLinputs)
InputsForOutputs

Collect the inputs that are used to construct the numerical and the graphical outputs

Description

Collect the inputs that are used to construct the numerical and the graphical outputs

Usage

InputsForOutputs(
    ModelType,
    Horiz,
    ListOutputWished,
    OutputLabel,
    WishStationarityQ,
    UnitYields,
    WishGraphYields = 0,
    WishGraphRiskFactors = 0,
    WishOrthoJLLgraphs = 0,
    WishForwardPremia = 0,
    LimFP = NULL,
    WishBootstrap = 0,
    ListBoot = NULL,
    WishForecast = 0,
    ListForecast = NULL
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModelType</td>
<td>String-vector containing the label of the model to be estimated</td>
</tr>
<tr>
<td>Horiz</td>
<td>Single numerical vector containing the desired horizon of analysis for the outputs</td>
</tr>
<tr>
<td>ListOutputWished</td>
<td>List of desired graphical outputs. Available options are: &quot;Fit&quot;, &quot;IRF&quot;, &quot;FEVD&quot;, &quot;GIRF&quot;, &quot;GFEVD&quot;.</td>
</tr>
<tr>
<td>OutputLabel</td>
<td>Name of the output label to be stored</td>
</tr>
<tr>
<td>WishStationarityQ</td>
<td>User must set 1 if she wishes to impose the largest eigenvalue under the Q to be strictly smaller than 1, otherwise set 0.</td>
</tr>
</tbody>
</table>
InputsForOutputs

UnitYields  (i) "Month": if maturity of yields are expressed in months or (ii) "Year": if maturity of yields are expressed in years

WishGraphYields
Binary variable: set 1, if the user wishes graphs to be generated; or set 0, otherwise. Default is set as "0".

WishGraphRiskFactors
Binary variable: set 1, if the user wishes graphs to be generated; or set 0, otherwise. Default is set as "0".

WishOrthoJLLgraphs
Binary variable: set 1, if the user wishes orthogonalized JLL-based graphs to be generated; or set 0, otherwise. Default is set as "0".

WishForwardPremia
Binary variable: set 1, if the user wishes graphs to be generated; or set 0, otherwise. Default is set as "0".

LimFP
Numerical vector containing the maturities associated with the starting and the ending date of the loan

WishBootstrap
Binary variable: set 1, if the user wishes graphs to be generated; or set 0, otherwise. Default is set as "0".

ListBoot
List containing the four following elements:
1. "methodBS": Desired bootstrap method among (a) 'bs' (standard residual bootstrap), (b) 'wild' (wild bootstrap), (c) 'block' (block bootstrap);
2. "BlockLength": if block bootstrap is chosen, then the user has to specify the length of the block (single numerical vector);
3. "ndraws": number of draws;
4. "pctg": level of confidence (single numerical vector expressed in basis points)

WishForecast
Binary variable: set 1, if the user wishes graphs to be generated; or set 0, otherwise. Default is set as "0".

ListForecast
List containing the three following elements:
1. "ForHoriz": forecast horizon;
2. "t0Sample": index of the first variable of the information set;
3. "t0Forecast": index of the first forecast cut-off date.

Value
List of necessary inputs to generate the graphs of the outputs of the desired model

Examples

```r
ModelType <- "JPS"
Horiz <- 100
DesiredOutputGraphs <- c("Fit", "GIRF", "GFEVD")
OutputLabel <- "Test"
WishStationarityQ <- 1
WishGraphRiskFac <- 0
```
$\text{WishGraphYields} \leftarrow 1$

$\text{InputsList} \leftarrow \text{InputsForOutputs}(\text{ModelType}, \text{Horiz}, \text{DesiredOutputGraphs}, \text{OutputLabel},$

$\text{WishStationarityQ}, \text{WishGraphYields}, \text{WishGraphRiskFac})$

---

**JLL**

*Set of inputs present at JLL's P-dynamics*

---

**Description**

Set of inputs present at JLL's P-dynamics

**Usage**

$\text{JLL}(\text{NonOrthoFactors}, N, \text{JLLinputs})$

**Arguments**

- **NonOrthoFactors**
  - Risk factors before the orthogonalization (FxT)
- **N**
  - Number of country-specific spanned factors
- **JLLinputs**
  - List of necessary inputs to estimate JLL outputs:
    1. Economies: set of economies that are part of the economic system (string-vector)
    2. "DomUnit": name of the economy which is assigned as the dominant unit.
       If no dominant unit is assigned, then this variable is defined as "None"
    3. WishSigmas: equal to "1" if one wishes the variance-covariance matrices
       and the Cholesky factorizations (can take long if they need to be estimated).
       Set "0", otherwise.
    4. SigmaNonOrtho: NULL or some F x F matrix from the non-orthogonalized dynamics
    5. JLLModelType: available options are "JLL original", "JLL jointSigma" or
       "JLL NoDomUnit"

**Details**

For the models 'JLL original' or "JLL jointSigma" the name of one dominant economy must as-
signed.

For the model 'JLL NoDomUnit', the name of one dominant economy must be set as "None".

**Value**

List of model parameters from both the orthogonalized and non-orthogonalized versions of the
JLL's based models
References

Examples

data(CM_Factors)
ZZ <- RiskFactors
N <- 3

JLLinputs <- list()
JLLinputs$Economies <- c("China", "Brazil", "Mexico", "Uruguay")
JLLinputs$DomUnit <- "China"
JLLinputs$WishSigmas <- 1
JLLinputs$SigmaNonOrtho <- NULL
JLLinputs$JLLModelType <- "JLL original"

JLL(ZZ, N, JLLinputs)

K1XQStationary

Impose stationarity under the Q-measure

Description
Impose stationarity under the Q-measure

Usage
K1XQStationary(StationaryEigenvalues)

Arguments
StationaryEigenvalues
Binary variable: set "1" if the user whises the largest eigenvalue to be strictly smaller than 1. Set "0", otherwise

Value
list

Examples
stat <- 1 # Takes values 1 and 0
K1XQStationary(stat)
LabFac

Generates the labels factors

Description
Generates the labels factors

Usage
LabFac(N, DomVar, GlobalVar, Economies, ModelType)

Arguments
N number of spanned factors per country (scalar)
DomVar character-vector containing the names of the domestic variables
GlobalVar character-vector containing the names of the global variables
Economies string-vector containing the names of the economies which are part of the economic system
ModelType string-vector containing the label of the model to be estimated

Value
List containing the country-specific risk factor labels

Examples
N <- 2
DomVar <- c("inflation", "Economic growth")
GlobalVar <- "Commodity Prices"
Economies <- c("U.S.", "Canada", "Germany", "Japan")
ModelType <- "JPS"

VarLabels <- LabFac(N, DomVar, GlobalVar, Economies, ModelType)

ListModelInputs

Concatenate the model-specific inputs in a list

Description
Concatenate the model-specific inputs in a list
Usage

ListModelInputs(
    ModelType,
    Data = NULL,
    Economies,
    VARXtype = NULL,
    t_First_Wgvar = NULL,
    t_Last_Wgvar = NULL,
    W_type = NULL,
    DomUnit = NULL,
    WishSigmas = NULL,
    SigmaNonOrtho = NULL,
    BiasCorrection = 0,
    flag_mean = NULL,
    gamma = NULL,
    N_iter = NULL,
    N_burn = NULL,
    B = NULL,
    checkBRW = NULL,
    B_check = NULL,
    DataPathTrade = NULL
)

Arguments

ModelType  string-vector containing the label of the model to be estimated
Data       dataset generated from the "DataForEstimation" function
Economies  string-vector containing the names of the economies of the system
VARXtype   string-vector containing the VARX feature (see "GVAR" function) (GVAR-based models)
t_First_Wgvar Sample starting date (year) (GVAR-based models)
t_Last_Wgvar Sample last date (year) (GVAR-based models)
W_type     Criterion used in the computation of the star variables (see "Transition_Matrix" function) (GVAR-based models)
DomUnit    name of the economy which is assigned as the dominant unit (JLL-based models)
WishSigmas equal to "1" if one wishes the variance-covariance matrices and the Cholesky factorizations (JLL-based models)
SigmaNonOrtho NULL or some F x F matrix from the non-orthogonalized dynamics (JLL-based models)
BiasCorrection binary variable. it takes value equal to 1 if the user wishes the estimates to be bias-corrected and 0, otherwise. (BRW model)
flag_mean  flag whether mean- (TRUE) or median- (FALSE) unbiased estimation is desired
gamma      adjustment parameter (BRW model)
**Maturities**

- **N_iter**: number of iterations (BRW model)
- **N_burn**: number of burn-in iterations (BRW model)
- **B**: number of bootstrap samples (BRW model)
- **checkBRW**: flag whether the user wishes to perform the closeness check (BRW model)
- **B_check**: number of bootstrap samples for closeness check
- **DataPathTrade**: path of the Excel file containing the data (if any)

**Examples**

```r
ModelType <- "JLL original"
Eco <- c("China", "Brazil", "Mexico", "Uruguay")
DU <- "China"
Sig <- 1
NonOrtho <- 0

ListModelInputs(ModelType, Economies = Eco, DomUnit = DU, WishSigmas = Sig, SigmaNonOrtho = NonOrtho)
```

---

**Description**

Create a vector of numerical maturities in years

**Usage**

```r
Maturities(DataYields, Economies, UnitYields)
```

**Arguments**

- **DataYields**: matrix containing all yields of the system (JxT, if the model is single-country-based or CJxT if the model is multy-country-based)
- **Economies**: vector containing names of all the economies of the system
- **UnitYields**: (i) "Month": if maturity of yields are expressed in months or (ii) "Year": if maturity of yields are expressed in years

**Value**

Vector containing all observed maturities expressed in years

**Examples**

```r
data('CM_Yields')
Economies <- c("China", "Brazil", "Mexico", "Uruguay")
Maturities(Yields, Economies, "Month")
```
ModelPara

Replications of the JPS (2014) outputs by the MultiATSM package

Description

Unspanned macro risk model outputs by the MultiATSM package

Usage

data("JPSrep")

Format

list of inputs and outputs

inputs  general model inputs
ests    model parameters estimates (JPS form)
llk     log-likelihood of the observations
rot     model parameters estimates (rotation form)

MultiATSM

ATSM Package

Description

Estimation of several classes of affine term structure of interest rates models.

Author(s)

Rubens Moura <rubens.gtmoura@gmail.com>
### NumOutputs

**Construct the model numerical outputs (model fit, IRFs, GIRFs, FEVDs, GFEVDs, and risk premia decomposition)**

**Description**

Construct the model numerical outputs (model fit, IRFs, GIRFs, FEVDs, GFEVDs, and risk premia decomposition)

**Usage**

```
NumOutputs(ModelType, ModelPara, InputsForOutputs, FactorLabels, Economies)
```

**Arguments**

- **ModelType**: a string-vector containing the label of the model to be estimated
- **ModelPara**: List of model parameter estimates (See the "Optimization" function)
- **InputsForOutputs**: list containing the desired horizon of analysis for the model fit, IRFs, GIRFs, FEVDs, GFEVDs and risk premia decomposition
- **FactorLabels**: a string-list based which contains all the labels of all the variables present in the model
- **Economies**: a string-vector containing the names of the economies which are part of the economic system

**Value**

List of the model numerical outputs, namely

1. Model fit of bond yields
2. IRFs
3. FEVDs
4. GIRFs
5. GFEVDs
6. Risk premia decomposition

**Examples**

```
# See examples in the vignette file of this package (Section 4).
```
Optimization

Perform the minimization of mean(f)

Description

Perform the minimization of mean(f)

Usage

Optimization(
    f,
    tol,
    varargin,
    FactorLabels,
    Economies,
    ModelType,
    JLLinputs = NULL,
    GVARinputs = NULL
)

Arguments

f                  vector-valued objective function (function)
 tol               convergence tolerance (scalar). For ML estimation, a reasonable value is tol <- 1e-4
 varargin           list containing starting values and constraints: for each input argument K (of f), we need four inputs that look like:
                       1. a starting value: K0
                       2. a variable label ('K0') followed by a ':' followed by a type of constraint. The constraint can be:
                           • 'bounded': bounded matrix;
                           • 'Jordan' or 'Jordan MultiCountry': a matrix of Jordan type;
                           • 'psd': psd matrix;
                           • 'stationary': largest eigenvalue of the risk-neutral feedback matrix is strictly smaller than 1;
                           • 'diag' or 'BlockDiag': a diagonal or block diagonal matrix.
                           • 'JLLstructure': to impose the zero-restrictions on the variance-voriance matrix along the lines of the JLL models
                       3. a lower bound lb (lb <- NULL -> no lower bound)
                       4. an upper bound ub (ub <- NULL -> no upper bound)
                       5. Specification of the optimization settings:
                           • 'iter off': hide the printouts of the numerical optimization routines;
                           • 'fminunc only': only uses fminunc for the optimization;
                           • 'fminsearch only': only uses fminsearch for the optimization.
### Description
Create the variable labels used in the estimation

### Usage
```
ParaLabels(ModelType, WishStationarityQ)
```

### Arguments
- **ModelType**: a string-vector containing the label of the model to be estimated
- **WishStationarityQ**: User must set "1" if she wishes to impose the largest eigenvalue under the Q to be strictly smaller than 1. Otherwise set "0"
Value

list containing the features of the parameters that will be used in the estimation

Examples

```r
ModelType <- "GVAR jointQ"
WishStationarityQ <- 1
ParaLabels(ModelType, WishStationarityQ)
```

---

**pca_weights_one_country**

*Weight matrix from principal components (matrix of eigenvectors)*

Description

Weight matrix from principal components (matrix of eigenvectors)

Usage

```r
pca_weights_one_country(Y, Economy)
```

Arguments

- **Y**
  - matrix dimension (J x T), where J - the number of maturities and T - time series length
- **Economy**
  - string-vector containing the name of one economy

Value

matrix (J x J)

Examples

```r
data("CM_Yields")
pca_weights_one_country(Yields, Economy= "Brazil")
```
Estimate the risk-neutral feedback matrix $K_{1Q}$ using linear regressions

**Description**

Estimate the risk-neutral feedback matrix $K_{1Q}$ using linear regressions

**Usage**

```r
Reg_K1Q(Y, mat, Z, dt, type)
```

**Arguments**

- **Y**: matrix of yields used in estimation ($J \times T$)
- **mat**: vector of maturities (in years) of yields used in estimation ($J \times 1$)
- **Z**: pricing factors (can be yields-based or non-yields/macro variables) ($N \times T$)
- **dt**: time unit of the model (scalar). For instance, if data is (i) monthly, $dt < - 12$; (ii) quarterly, $dt < - 4$; (iii) yearly, $dt <- 1$.
- **type**: 'Jordan' -> $K_{1Q}$ will be of the Jordan type

**Value**

Risk neutral feedback matrix $K_{1Q}$.

**References**

This function is based on the "Reg_K1Q" function by Le and Singleton (2018). "A Small Package of Matlab Routines for the Estimation of Some Term Structure Models." (Euro Area Business Cycle Network Training School - Term Structure Modelling). Available at: https://cepr.org/40029

**Examples**

```r
data(CM_Yields)
Y_China <- Yields[1:6,]
Z_China <- Spanned_Factors(Y_China, Economies ="China", N=3)
mat <-c(0.25 , 0.5 , 1, 3, 5, 10)
dt <- 1/12
type <- 'Jordan'
Reg_K1Q(Y_China, mat, Z_China, dt, type)
```
### RiskFactors

**Data:** Risk Factors - Candelon and Moura (2021)

#### Description

Risk factors data used in Candelon and Moura (2021)
Risk factors data used in Candelon and Moura (2023)

#### Usage

```r
data("CM_Factors")
data("CM_Factors_2023")
```

#### Format

- matrix containing the risk factors of the models
- matrix containing the risk factors of the models

#### References

Candelon, B. and Moura, R. "A Multi-Country Model of the Term Structures of Interest Rates with a GVAR".
Candelon, B. and Moura, R. "Sovereign yield curves and the COVID-19 in emerging markets".

### RMSEsep

**Compute the root mean square error ("sep Q" models)**

#### Description

Compute the root mean square error ("sep Q" models)

#### Usage

```r
RMSEsep(ForecastOutputs)
```

#### Arguments

- `ForecastOutputs`
  List of country-specific forecasts (see "ForecastYieldsSepQ" function)
Spanned_Factors

*Compute the country-specific spanned factors*

**Description**

Compute the country-specific spanned factors

**Usage**

```
Spanned_Factors(Yields, Economies, N)
```

**Arguments**

- **Yields**: matrix (J x T), where J - the number of maturities and T - time series length
- **Economies**: C-dimensional string-vector containing the names of the economies which are part of the economic system
- **N**: scalar: desired number of spanned factors (maximum number allowed is N = J)

**Value**

Matrix containing the N spanned for all the countries of the system (CJ x T)

**Examples**

```
data(CM_Yields)
Economies <- c("China", "Brazil", "Mexico", "Uruguay")
N <- 3
Spanned_Factors(Yields, Economies, N)
```

---

StarFactors

*Generates the star variables necessary for the GVAR estimation*

**Description**

Generates the star variables necessary for the GVAR estimation

**Usage**

```
StarFactors(RiskFactors, Economies, W)
```

**Arguments**

- **RiskFactors**: time series of the risk factors (F x T)
- **Economies**: string-vector containing the names of the economies which are part of the economic system
- **W**: GVAR transition matrix (C x C)
Trade Flows

Value

List containing the star factors of each country of the economic system

Examples

data(CM_Factors)
Economies <- c("China", "Brazil", "Mexico", "Uruguay")
Wgvar <- matrix(c(0, 0.83, 0.86, 0.38, 0.65, 0, 0.13, 0.55,
                   0.32, 0.12, 0, 0.07, 0.03, 0.05, 0.01, 0), nrow = 4, ncol = 4)
rownames(Wgvar) <- Economies
colnames(Wgvar) <- Economies
StarFactors(RiskFactors, Economies, Wgvar)

Trade Flows  Data: Trade Flows - Candelon and Moura (2021)

Description

Trade Flows data used in Candelon and Moura (2021)

Usage

data("CM_Trade")

Format

list containing the bilateral trade flows

References

Candelon, B. and Moura, R. "A Multi-Country Model of the Term Structures of Interest Rates with a GVAR".

Trade Flows  Data: Trade Flows - Candelon and Moura (2023)

Description

Trade Flows data used in Candelon and Moura (2023)

Usage

data("CM_Trade")
**Transition_Matrix**

**Format**
list containing the bilateral trade flows

**References**
Candelon, B. and Moura, R. "Sovereign yield curves and the COVID-19 in emerging markets".

---

**Transition_Matrix**  
*Compute the transition matrix required in the estimation of the GVAR model*

**Description**
Compute the transition matrix required in the estimation of the GVAR model

**Usage**

```r
Transition_Matrix(
  t_First,
  t_Last,
  Economies,
  type,
  DataPath = NULL,
  Data = NULL
)
```

**Arguments**
- **t_First**: Sample starting date (year)
- **t_Last**: Sample last date (year)
- **Economies**: Vector containing the names of all the economies of the system.
- **type**: Three possibilities:
  - "Full Sample": if one wishes ALL weight matrices of each year from which data is available (it may extrapolate the sample period);
  - "Sample Mean": if one wishes a SINGLE weight matrix containing the average of weights over of the entire sample period;
  - Some year in particular (e.g. "1998", "2005" ...).
- **DataPath**: path of the Excel file containing the data (if any). The default is linked to the Excel file available in the package.
- **Data**: Data for computing the transition matrix. Default is set to NULL.

**Details**
NOTE: if there is missing data for any country of the system for that particularly year, then the transition matrix will include only NAs.
Value

matrix or list of matrices

Examples

data(CM_Trade)

    t_First <- "2006"
    t_Last <- "2019"
    Economies <- c("China", "Brazil", "Mexico", "Uruguay")
    type <- "Sample Mean"
    Transition_Matrix(t_First, t_Last, Economies, type, DataPath = NULL, Data = TradeFlows)

---

VAR

Estimates a VAR(1)

Description

Estimates a VAR(1)

Usage

VAR(RiskFactors, VARtype, Bcon = NULL)

Arguments

RiskFactors  matrix containing all the risk factors (K x T)
VARtype      string-vector which accommodates two possibilities: 'unconstrained' or 'constrained'
Bcon         constraints matrix (K+1 x N) - should contain an intercept. If Bcon(i,j) = nan -> B(i,j) is a free parameter. Default is set to NULL.

Value

intercept, feedback matrix and the variance-covariance matrix of a VAR(1)

Examples

data("CM_Factors")
#Example 1
VAR(RiskFactors, VARtype = 'unconstrained')
#Example 2
K <- nrow(RiskFactors)
Bcon <- matrix(0, nrow = K, ncol = K+1)
Bcon[,1:3] <- NaN
Yields

\[ \text{VAR(RiskFactors, VARtype='constrained', Bcon)} \]

---

<table>
<thead>
<tr>
<th>Yields</th>
<th>Data: Yields - Candelon and Moura (2021)</th>
</tr>
</thead>
</table>

**Description**

Yields data used in Candelon and Moura (2021)

Bond yield data used in Candelon and Moura (2023)

**Usage**

```r
data("CM_Yields")

data("CM_Yields_2023")
```

**Format**

- matrix containing the Yields of the models
- matrix containing the Yields of the models

**References**

Candelon, B. and Moura, R. "A Multi-Country Model of the Term Structures of Interest Rates with a GVAR".

Candelon, B. and Moura, R. "Sovereign yield curves and the COVID-19 in emerging markets".
Index

* (2014)
  - BR_jps_out, 6
  - ModelPara, 24
* Factors
  - FactorsGVAR, 10
  - GVARFactors, 14
  - RiskFactors, 30
* Flows
  - Trade_Flows, 32
  - TradeFlows, 32
* GVAR
  - FactorsGVAR, 10
  - GVARFactors, 14
* JPS
  - BR_jps_out, 6
  - ModelPara, 24
* Risk
  - FactorsGVAR, 10
  - GVARFactors, 14
  - RiskFactors, 30
* Trade
  - Trade_Flows, 32
  - TradeFlows, 32
* Yields
  - Yields, 35
* outputs
  - BR_jps_out, 6
  - ModelPara, 24
Bias_Correc_VAR, 2
Bootstrap, 5
BR_jps_out, 6
DatabasePrep, 7
DataForEstimation, 8
FactorsGVAR, 10
ForecastYields, 10
Functionf, 11
GVAR, 12
GVARFactors, 14
InputsForMLEdensity, 14
InputsForOutputs, 17
JLL, 19
K1XQStationary, 20
LabFac, 21
ListModelInputs, 21
Maturities, 23
ModelPara, 24
MultiATSM, 24
MultiATSM-package (MultiATSM), 24
NumOutputs, 25
Optimization, 26
ParaLabels, 27
pca_weights_one_country, 28
Reg_K1Q, 29
RiskFactors, 30
RMSEsep, 30
Spanned_Factors, 31
StarFactors, 31
Trade_Flows, 32
TradeFlows, 32
Transition_Matrix, 33
VAR, 34
Yields, 35