Package 'MultiATSM'

September 2, 2025

Type Package

Title Multicountry Term Structure of Interest Rates Models

Version 1.4.0

Date 2025-09-02

Description Estimation routines for several classes of affine term structure of interest rates models. All the models are based on the single-country unspanned macroeconomic risk framework from Joslin, Priebsch, and Singleton (2014, JF) <doi:10.1111/jofi.12131>. Multicountry extensions such as the ones of Jotikasthira, Le, and Lundblad (2015, JFE) <doi:10.1016/j.jfineco.2014.09.004>, Candelon and Moura (2023, EM) <doi:10.1016/j.econmod.2023.106453>, and Candelon and Moura (2024, JFEC) <doi:10.1093/jjfinec/nbae008> are also available.

License GPL-2 | GPL-3

Encoding UTF-8

RoxygenNote 7.2.3

Imports cowplot, ggplot2, hablar, magic, neldermead, pracma

Suggests readxl, knitr, rmarkdown, bookdown, kableExtra, magrittr

Depends R (>= 4.3.0)

VignetteBuilder knitr

URL https://github.com/rubensmoura87/MultiATSM

BugReports https://github.com/rubensmoura87/MultiATSM/issues

NeedsCompilation no

Author Rubens Moura [aut, cre] (ORCID:
<https://orcid.org/0000-0001-8105-4729>)

Maintainer Rubens Moura <rubens.gtmoura@gmail.com>

Repository CRAN

Date/Publication 2025-09-02 03:20:02 UTC

Contents

autoplot . . . 3
autoplot.ATSMModelBoot . . . 3
autoplot.ATSMNumOutputs . . . 4
Bias_Correc_VAR . . . 4
Bootstrap . . . 6
BR_jps_out . . . 8
DatabasePrep . . . 8
DataForEstimation . . . 10
DomesticMacroVar . . . 12
DomMacro . . . 12
FactorsGVAR . . . 13
FEVDandGFEVDgraphs . . . 13
Fitgraphs . . . 15
ForecastYields . . . 16
GlobalMacro . . . 17
GlobalMacroVar . . . 18
GVAR . . . 18
InpForOutEx . . . 20
InputsForOpt . . . 20
InputsForOutputs . . . 23
IRFandGIRFgraphs . . . 25
JLL . . . 26
LabFac . . . 28
LoadData . . . 28
ModelPara . . . 29
ModelParaEx . . . 30
MultiATSM . . . 30
NumOutEx . . . 31
NumOutputs . . . 31
Optimization . . . 33
Out . . . 34
pca_weights_one_country . . . 35
plot.ATSMModelForecast . . . 36
print.ATSMModelInputs . . . 36
RiskFactors . . . 37
RiskFactorsGraphs . . . 37
Spanned_Factors . . . 38
StarFactors . . . 39
summary.ATSMModelInputs . . . 39
summary.ATSMModelOutputs . . . 40
TPDecompGraph . . . 40
TradeFlows . . . 41
Trade_Flows . . . 42
Transition_Matrix . . . 42
VAR . . . 43
Yields . . . 44

Index 46

`autoplot` *Autoplot generic function*

Description

Autoplot generic function

Usage

```
autoplot(x, ...)
```

Arguments

`x`	Object to plot
`...`	Additional arguments passed to methods

`autoplot.ATSMModelBoot` *Autoplot method for ATSMModelBoot objects*

Description

Autoplot method for ATSMModelBoot objects

Usage

```
## S3 method for class 'ATSMModelBoot'
autoplot(x, NumOutPE, type, ...)
```

Arguments

`x`	An object of class 'ATSMModelBoot'
`NumOutPE`	An object of class 'ATSMNumOutputs': point estimates of the numerical outputs
`type`	Plot type: one of "IRF", "FEVD", "GIRF", "GFEVD" (each must be suffixed with "_Factors" or "_Yields"). For JLL-based models, an additional "_Ortho" suffix produces orthogonalized outputs. All inputs must end by "_Boot" as a reference to the bootstrap procedure.
`...`	Additional arguments (not used)

`autoplot.ATSMNumOutputs` *Autoplot method for ATSMNumOutputs objects*

Description

Autoplot method for ATSMNumOutputs objects

Usage

```
## S3 method for class 'ATSMNumOutputs'
autoplot(x, type, ...)
```

Arguments

`x`	An object of class 'ATSMNumOutputs'
`type`	Plot type: "RiskFactors", "Fit", "TermPremia", or one of "IRF", "FEVD", "GIRF", "GFEVD" (each must be suffixed with "_Factors" or "_Yields"). For JLL-based models, an additional "_Ortho" suffix produces orthogonalized outputs.
`...`	Additional arguments (not used)

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

Description

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

Usage

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

Arguments

`ModelType`	A character vector indicating the model type to be estimated.
`BRWinputs`	A list containing the necessary inputs for the BRW model estimation: 1. `flag_mean`: Logical. Determines whether mean- (TRUE) or median- (FALSE) unbiased estimation is desired. Default is TRUE. 2. `gamma`: Numeric. Adjustment parameter between 0 and 1. Default is 0.5. 3. `N_iter`: Integer. Number of iterations for the stochastic approximation algorithm after burn-in. Default is 5,000. 4. `N_burn`: Integer. Number of burn-in iterations. Default is 15 5. `B`: Integer. Number of bootstrap samples per iteration for calculating the noisy measure of the OLS estimator's mean or median. Default is 50. 6. `check`: Logical. Indicates whether to perform a closeness check. Default is TRUE. 7. `B_check`: Integer. Number of bootstrap samples for the closeness check. Default is 100,000.
`RiskFactors`	A numeric matrix (T x F) representing the time series of risk factors.
`N`	Integer. Number of country-specific spanned factors.
`Economies`	A character vector containing the names of the economies included in the system.
`FactorLabels`	A list of character vectors with labels for all variables in the model.
`GVARinputs`	List. Inputs for GVAR model estimation (see `GVAR` function). Default is NULL.
`JLLinputs`	List. Inputs for JLL model estimation (see `JLL` function). Default is NULL.
`ev_restr`	Numeric. Restriction on the largest eigenvalue under the P-measure. Default is 1.
`nargout`	Integer. Number of elements in the output list. Default is 4.
`verbose`	verbose Logical flag controlling function messaging. Default is TRUE.

Value

Bias-corrected VAR parameters based on the framework of Bauer, Rudebusch and Wu (2012). The list contains:

1. `Phi_tilde`: estimated coefficient matrix (F x F);
2. `mu_tilde`: estimated intercept (F x 1);
3. `V_tilde`: estimated variance-covariance matrix (F x F);
4. `dist`: root mean square distance (scalar);
5. `Phi_sample`: sample estimated variance-covariance matrix used in the checks (F x F x B_check) - this output is reported if nargout is 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(flag_mean = TRUE, gamma = 0.4, N_iter = 1000, N_burn = 100,
                  B = 10, check = 1, B_check = 5000)

Economies <- "China"
N <- 3
ModelType <- "JPS original"
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,
  Economies,
  InputsForOutputs,
  FactorLabels,
  JLLlist,
  GVARlist,
  WishBC,
  BRWlist,
  Folder2save = NULL,
  verbose = TRUE
)
```

Arguments

`ModelType`	A character vector indicating the model type to be estimated.
`ModelParaPE`	A list containing the point estimates of the model parameters. For details, refer to the outputs from the `Optimization` function.
`NumOutPE`	The point estimate derived from numerical outputs. See the outputs from the `NumOutputs` function for further information.

Economies	A character vector containing the names of the economies included in the system.
InputsForOutputs	A list containing the necessary inputs for generating IRFs, GIRFs, FEVDs, GFEVDs and Term Premia.
FactorLabels	A list of character vectors with labels for all variables in the model.
JLLlist	List. Inputs for JLL model estimation (see JLL function). Default is NULL.
GVARlist	List. Inputs for GVAR model estimation (see GVAR function). Default is NULL.
WishBC	Whether to estimate the physical parameter model with bias correction, based on the method by Bauer, Rudebusch and Wu (2012) (see Bias_Correc_VAR function). Default is set to 0.
BRWlist	List of necessary inputs for performing the bias-corrected estimation (see Bias_Correc_VAR function).
Folder2save	Folder path where the outputs will be stored. Default option saves the outputs in a temporary directory.
verbose	Logical flag controlling function messaging. Default is TRUE.

Value

An object of class 'ATSMModelBoot' containing the following keys elements:

- List of model parameters for each draw
- List of numerical outputs (IRFs, GIRFs, FEVDs and GFEVDs) for each draw
- Confidence bounds for the chosen level of significance

Examples

```
data("ParaSetEx")
data("InpForOutEx")
data("NumOutEx")
ModelType <- "JPS original"
Economy <- "Brazil"
FacLab <- LabFac(N = 1, DomVar = "Eco_Act", GlobalVar = "Gl_Eco_Act", Economy, ModelType)

# Adjust Forecasting setting
InpForOutEx[[ModelType]]$Bootstrap <- list(WishBootstrap = 1, methodBS = 'bs', BlockLength = 4,
                                           ndraws = 5, pctg =  95)

Boot <- Bootstrap(ModelType, ModelParaEx, NumOutEx, Economy, InpForOutEx, FacLab, JLLlist = NULL,
           GVARlist = NULL, WishBC = 0, BRWlist = NULL, Folder2save  = NULL, verbose = TRUE)
```

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

References

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

DatabasePrep — *Gather data of several countries in a list. Particularly useful for GVAR-based setups (Compute "GVARFactors")*

Description

Gather data of several countries in a list. Particularly useful for GVAR-based setups (Compute "GVARFactors")

Usage

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

Arguments

t_First	Start date of the sample period in the format yyyy-mm-dd.
t_Last	End date of the sample period in the format yyyy-mm-dd.
Economies	A character vector containing the names of the economies included in the system.
N	Integer. Number of country-specific spanned factors.
FactorLabels	A list of character vectors with labels for all variables in the model.
ModelType	A character vector indicating the model type to be estimated.
Wgvar	GVAR transition matrix of size C x C, applicable if a GVAR-type model is selected. Default is NULL.
DataPathMacro	File path to the Excel file containing macroeconomic data, if provided. The default path points to the Excel file available within the package.
DataPathYields	File path to the Excel file containing yields data, if provided. The default path points to the Excel file available within the package

Value

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

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 <- 3
ModelType <- "GVAR multi"
FactorLabels <-  LabFac(N, DomVar, GlobalVar, Economies, ModelType)
Wgvar <- Transition_Matrix(t_First = "2006", t_Last= "2019", Economies, type = "Sample Mean")


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

`DataForEstimation` *Retrieves data from Excel and build the database used in the model estimation*

Description

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

Usage

```
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`	Start date of the sample period in the format yyyy-mm-dd.
`tF`	End date of the sample period in the format yyyy-mm-dd.
`Economies`	A character vector containing the names of the economies included in the system.
`N`	Integer. Number of country-specific spanned factors.
`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
`DataFrequency`	Character-based-vector. Available options are: "Daily All Days", "Daily Business Days", "Weekly", "Monthly", "Quarterly", "Annually"
`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, J x T or CJ x T);
2. time series of the complete set risk factors (matrix, K x T);
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.

See Also

`InputsForOpt`

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 original"
FactorLabels <-  LabFac(N, DomVar, GlobalVar, Economies, ModelType)
DataFrequency <- "Monthly"

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

DomesticMacroVar *Data: Risk Factors - Candelon and Moura (2024, JFEC)*

Description

Risk factors data used in Candelon and Moura (2024, JFEC)

Usage

```
data("CM_DomMacroFactors")
```

Format

matrix containing the risk factors of the models

References

Candelon, B. and Moura, R. (2024) "A Multicountry Model of the Term Structures of Interest Rates with a GVAR". (Journal of Financial Econometrics)

DomMacro *Data: Risk Factors for the GVAR - Candelon and Moura (2023)*

Description

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

Usage

```
data("CM_DomMacro_2023")
```

Format

list containing the variables used in the GVAR models

References

Candelon, B. and Moura, R. (2023) "Sovereign yield curves and the COVID-19 in emerging markets". (Economic Modelling)

FactorsGVAR *Data: Risk Factors for the GVAR - Candelon and Moura (2024, JFEC)*

Description

Risk factors data used in the GVAR models - Candelon and Moura (2024, JFEC)

Usage

```
data("CM_Factors_GVAR")
```

Format

list containing the variables used in the GVAR models

References

Candelon, B. and Moura, R. (2024) "A Multicountry Model of the Term Structures of Interest Rates with a GVAR". (Journal of Financial Econometrics)

FEVDandGFEVDgraphs *FEVD and GFEVD graphs for all models*

Description

FEVD and GFEVD graphs for all models

Usage

```
FEVDandGFEVDgraphs(
  ModelType,
  NumOut,
  WishPdynamicsgraphs,
  WishYieldsgraphs,
  FEVDhoriz,
  PathsGraphs,
  OutputType,
  Economies,
  Folder2save,
  verbose
)
```

Arguments

ModelType	A character vector indicating the estimated model type
NumOut	list of computed outputs containing the model fit, IRFs, FEVDs, GIRFs, GFEVDs and Term premia
WishPdynamicsgraphs	binary variable: set 1, if the user wishes graphs to be generated; or set 0, otherwise
WishYieldsgraphs	binary variable: set 1, if the user wishes graphs to be generated; or set 0, otherwise
FEVDhoriz	single numerical vector containing the desired horizon of analysis for the FEVDs
PathsGraphs	Path of the folder in which the graphs will be saved
OutputType	Available options are 'FEVD' and 'GFEVD'
Economies	A character vector containing the names of the economies included in the system.
Folder2save	Folder path where the outputs will be stored.
verbose	Logical flag controlling function messaging.

Available Methods

- 'autoplot(object, type = "FEVD_Factor")', 'autoplot(object, type = "FEVD_Yields")', 'autoplot(object, type = "GFEVD_Yields")', 'autoplot(object, type = "GFEVD_Yields")'.

- For JLL-based models other option available are: 'autoplot(object, type = "FEVD_Factor-_Ortho")', 'autoplot(object, type = "FEVD_Yields_Ortho")', 'autoplot(object, type = "GFEVD_Yields_Ortho")', 'autoplot(object, type = "GFEVD_Yields_Ortho")'.

Examples

```
data("NumOutEx")
ModelType <- "JPS original"
Economy <- "Brazil"
FEVDhoriz <- 20

FEVDandGFEVDgraphs(ModelType, NumOutEx, WishPdynamicsgraphs = 0, WishYieldsgraphs= 1, FEVDhoriz,
                   PathsGraphs = NULL, OutputType = "FEVD", Economy,
                   Folder2save = NULL, verbose = FALSE)
```

`Fitgraphs` *Model fit graphs for all models*

Description

Model fit graphs for all models

Usage

```
Fitgraphs(
  ModelType,
  WishFitgraphs,
  ModelPara,
  NumOut,
  Economies,
  PathsGraphs,
  Folder2save,
  verbose
)
```

Arguments

`ModelType`	A character vector indicating the estimated model type
`WishFitgraphs`	binary variable: set 1, if the user wishes graphs to be generated; or set 0, otherwise
`ModelPara`	List of model parameter estimates (See the `Optimization` function)
`NumOut`	A list of outputs containing the model fit, IRFs, FEVDs, GIRFs, GFEVDs and Term premia
`Economies`	A character vector containing the names of the economies included in the system.
`PathsGraphs`	Path of the folder in which the graphs will be saved
`Folder2save`	Desired folder path to save outputs
`verbose`	Logical flag controlling function messaging

Available Methods

- ‘autoplot(object, type = "Fit")‘

Examples

```
data("ParaSetEx")
data("NumOutEx")
ModelType <- "JPS original"
Economy <- "Brazil"

Fitgraphs(ModelType, WishFitgraphs = 1, ModelParaEx, NumOutEx, Economy, PathsGraphs = NULL,
```

```
            Folder2save = NULL, verbose = FALSE)
```

`ForecastYields` *Generates forecasts of bond yields for all model types*

Description

Generates forecasts of bond yields for all model types

Usage

```
ForecastYields(
  ModelType,
  ModelPara,
  InputsForOutputs,
  FactorLabels,
  Economies,
  JLLlist = NULL,
  GVARlist = NULL,
  WishBRW,
  BRWlist = NULL,
  Folder2save = NULL,
  verbose = TRUE
)
```

Arguments

`ModelType`	A character vector indicating the model type to be estimated.
`ModelPara`	A list containing the point estimates of the model parameters. For details, refer to the outputs from the `Optimization` function.
`InputsForOutputs`	A list containing the necessary inputs for generating IRFs, GIRFs, FEVDs, GFEVDs and Term Premia.
`FactorLabels`	A list of character vectors with labels for all variables in the model.
`Economies`	A character vector containing the names of the economies included in the system.
`JLLlist`	A list of necessary inputs for the estimation of JLL-based models (see the `JLL` function).
`GVARlist`	A list containing the necessary inputs for the estimation of GVAR-based models (see the `GVAR` function).
`WishBRW`	Whether to estimate the physical parameter model with bias correction, based on the method by Bauer, Rudebusch and Wu (2012) (see `Bias_Correc_VAR` function). Default is set to 0.

BRWlist	List of necessary inputs for performing the bias-corrected estimation (see Bias_Correc_VAR function).
Folder2save	Folder path where the outputs will be stored. Default option saves the outputs in a temporary directory.
verbose	Logical flag controlling function messaging. Default is TRUE.

Value

An object of class 'ATSMModelForecast' 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

Available Methods

- 'plot(object)'

Examples

```
data("ParaSetEx")
data("InpForOutEx")
# Adjust inputs according to the loaded features
ModelType <- "JPS original"
Economy <- "Brazil"
FacLab <- LabFac(N = 1, DomVar = "Eco_Act", GlobalVar = "Gl_Eco_Act", Economy, ModelType)
InpForOutEx[[ModelType]]$Forecasting <- list(WishForecast = 1, ForHoriz = 12,  t0Sample = 1,
                                             t0Forecast = 143, ForType = "Expanding")

Forecast <- ForecastYields(ModelType, ModelParaEx, InpForOutEx, FacLab, Economy,
                           WishBRW = 0, verbose = TRUE)
```

GlobalMacro *Data: Risk Factors - Candelon and Moura (2023)*

Description

Risk factors data used in Candelon and Moura (2023)

Usage

```
data("CM_GlobalMacro_2023")
```

Format

matrix containing the risk factors of the models

References

Candelon, B. and Moura, R. (2023) "Sovereign yield curves and the COVID-19 in emerging markets". (Economic Modelling)

`GlobalMacroVar` *Data: Risk Factors - Candelon and Moura (2024, JFEC)*

Description

Risk factors data used in Candelon and Moura (2024, JFEC)

Usage

```
data("CM_GlobalMacroFactors")
```

Format

matrix containing the risk factors of the models

References

Candelon, B. and Moura, R. (2024) "A Multicountry Model of the Term Structures of Interest Rates with a GVAR". (Journal of Financial Econometrics)

`GVAR` *Estimates a GVAR(1) and a VARX(1,1,1) models*

Description

Estimates a GVAR(1) and a VARX(1,1,1) models

Usage

```
GVAR(GVARinputs, N, CheckInputs = FALSE)
```

Arguments

`GVARinputs` List of inputs for GVAR model estimation:

1. `Economies`: A character vector containing the names of the economies included in the system.
2. `GVARFactors`: A list of all variables used in the estimation of the VARX model
   (see e.g. `CM_Factors_GVAR` file for details);
3. `VARXtype`: A character vector with three possible options:

- 'unconstrained': model is estimated without constraints (each equation is estimated individually by ordinary least square);
- 'constrained: Spanned Factors': The model is estimated with the restriction that foreign pricing factors do NOT affect (i) domestic economic variables and (ii) domestic pricing factors.
  (Equations are estimated using restricted least squares)
- 'constrained : [factor_name]': The model is estimated with the restriction that the specified risk factor is influenced only by its own lagged values and the lagged values of its corresponding star variables.
  (Equations are estimated using restricted least squares.)

4. Wgvar: The GVAR transition matrix (C x C) used in the model solution.
   (See the output from the Transition_Matrix function.).

N — Integer. Number of country-specific spanned factors.

CheckInputs — A logical flag to indicate whether to perform a prior consistency check on the inputs provided in GVARinputs. The default is set to FALSE

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

Chudik and Pesaran, (2016). "Theory and Practice of GVAR modelling" (Journal of Economic Surveys)

Examples

```
data(CM_Factors_GVAR)

GVARinputs <- list( Economies = c("China", "Brazil", "Mexico", "Uruguay"),
                    GVARFactors = FactorsGVAR, 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)
```

```
N <- 3

GVARPara <- GVAR(GVARinputs, N)
```

`InpForOutEx` *Example of list inputs used in the construction of several model outputs*

Description

List of inputs of a JPS-based model for Brazilian data

Usage

```
data("InpForOutEx")
```

Format

list of inputs

`InputsForOpt` *Generates inputs necessary to build the likelihood function for the ATSM model*

Description

Generates inputs necessary to build the likelihood function for the ATSM model

Usage

```
InputsForOpt(
  InitialSampleDate,
  FinalSampleDate,
  ModelType,
  Yields,
  GlobalMacro,
  DomMacro,
  FactorLabels,
  Economies,
  DataFrequency,
  GVARlist = NULL,
  JLLlist = NULL,
  WishBRW = FALSE,
  BRWlist = NULL,
  UnitYields = "Month",
  CheckInputs = TRUE,
  BS_Adj = FALSE,
  verbose = TRUE
)
```

Arguments

`InitialSampleDate`	Start date of the sample period in the format "dd-mm-yyyy"
`FinalSampleDate`	End date of the sample period in the format "dd-mm-yyyy"
`ModelType`	A character vector indicating the model type to be estimated. Available options: "JPS original", "JPS global", "GVAR single", "JPS multi", "GVAR multi", "JLL original", "JLL No DomUnit", "JLL joint Sigma".
`Yields`	A numerical matrix with time series of yields (JxT or CJ x T)
`GlobalMacro`	A numerical matrix with time series of the global risk factors (G x T)
`DomMacro`	A numerical matrix with time series of the country-specific risk factors for all C countries (CM x T)
`FactorLabels`	A list of character vectors with labels for all variables in the model.
`Economies`	A character vector containing the names of the economies included in the system.
`DataFrequency`	A character vector specifying the frequency of the data. Available options are: "Daily All Days", "Daily Business Days", "Weekly", "Monthly", "Quarterly", or "Annually".
`GVARlist`	A list containing the necessary inputs for the estimation of GVAR-based models
`JLLlist`	A list of necessary inputs for the estimation of JLL-based models. If the chosen model is "JLL original" or "JLL joint Sigma", then a dominant unit economy must be chosen. Otherwise, this list must be set as 'None'.
`WishBRW`	Logical. Whether to estimate the physical parameter model with bias correction, based on the method by Bauer, Rudebusch and Wu (2012). Default is FALSE.
`BRWlist`	List of necessary inputs for performing the bias-corrected estimation.
`UnitYields`	A character string indicating the maturity unit of yields. Options are: "Month" for yields expressed in months, or "Year" for yields expressed in years. Default is "Month".
`CheckInputs`	Logical. Whether to perform a prior check on the consistency of the provided input list. Default is TRUE.
`BS_Adj`	Logical. Whether to adjust the global series for the sepQ models in the Bootstrap setting. Default is FALSE.
`verbose`	Logical flag controlling function messaging. Default is TRUE.

Value

An object of class 'ATSMModelInputs' containing the necessary inputs for performing the model optimization.

Available Methods

- 'print(object)' - 'summary(object)'

Examples

```
# Example 1:
data(CM_GlobalMacroFactors)
data(CM_DomMacroFactors)
data(CM_Yields)

ModelType <- "JPS original"
Economies <- "Mexico"
t0 <- "01-05-2007" # Initial Sample Date (Format: "dd-mm-yyyy")
tF <- "01-12-2018" # Final Sample Date (Format: "dd-mm-yyyy")
N <- 3
GlobalVar <- c("Gl_Eco_Act") # Global Variables
DomVar <- c("Eco_Act") # Domestic Variables
FactorLabels <- LabFac(N, DomVar, GlobalVar, Economies, ModelType)

DataFreq <- "Monthly"

ATSMInputs <- InputsForOpt(t0, tF, ModelType, Yields, GlobalMacroVar, DomesticMacroVar,
                           FactorLabels, Economies, DataFreq, CheckInputs = FALSE)

# Example 2:
LoadData("CM_2024")

ModelType <- "GVAR multi"

Economies <- c("China", "Brazil", "Mexico", "Uruguay")
t0 <- "01-05-2007" # InitialSampleDate (Format: "dd-mm-yyyy")
tF <- "01-12-2019" # FinalSampleDate (Format: "dd-mm-yyyy")
N <- 2
GlobalVar <- c("Gl_Eco_Act", "Gl_Inflation") # Global Variables
DomVar <- c("Inflation") # Domestic Variables
FactorLabels <- LabFac(N, DomVar, GlobalVar, Economies, ModelType)

DataFreq <- "Monthly"
GVARlist <- list(VARXtype = "unconstrained", W_type = "Sample Mean",
                 t_First_Wgvar = "2007", t_Last_Wgvar = "2019")

ATSMInputs <- InputsForOpt(t0, tF, ModelType, Yields, GlobalMacroVar, DomesticMacroVar,
                       FactorLabels, Economies, DataFreq, GVARlist, CheckInputs = FALSE)

# Example 3:
if (requireNamespace('neldermead', quietly = TRUE)) {
LoadData("CM_2024")

ModelType <- "JLL original"

Economies <- c("China", "Brazil", "Uruguay")
t0 <- "01-05-2007" # InitialSampleDate (Format: "dd-mm-yyyy")
tF <- "01-12-2019" # FinalSampleDate (Format: "dd-mm-yyyy")
N <- 2
GlobalVar <- c("Gl_Eco_Act", "Gl_Inflation") # Global Variables
DomVar <- c("Eco_Act", "Inflation") # Domestic Variables
```

```
FactorLabels <- LabFac(N, DomVar, GlobalVar, Economies, ModelType)

JLLinputs <- list(DomUnit = "China")

DataFrequency <- "Monthly"

ATSMInputs <- InputsForOpt(t0, tF, ModelType, Yields, GlobalMacroVar, DomesticMacroVar,
                           FactorLabels, Economies, DataFreq, JLLlist = JLLinputs,
                           CheckInputs = FALSE)
} else {
 message("skipping functionality due to missing Suggested dependency")
}
```

`InputsForOutputs` *Collects the inputs that are used to construct the numerical and the graphical outputs*

Description

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

Usage

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

Arguments

`ModelType`	A character vector indicating the model type to be estimated.
`Horiz`	A numeric scalar specifying the desired analysis horizon for the outputs.

`ListOutputWished`
: A list of desired graphical outputs. Available options are: "Fit", "IRF", "FEVD", "GIRF", "GFEVD", "TermPremia".

`OutputLabel`
: A string for the name of the output label to be stored.

`WishStationarityQ`
: A binary variable (1 or 0) indicating whether to impose that the largest eigenvalue under Q is strictly smaller than 1. Set to 1 to impose the restriction, or 0 otherwise.

`DataFrequency`
: A character vector specifying the data frequency. Available options: "Daily All Days", "Daily Business Days", "Weekly", "Monthly", "Quarterly", "Annually".

`WishGraphYields`
: A binary variable (1 or 0) indicating whether the user wishes to generate graphs for yields. Default is 0.

`WishGraphRiskFactors`
: A binary variable (1 or 0) indicating whether the user wishes to generate graphs for risk factors. Default is 0.

`WishOrthoJLLgraphs`
: A binary variable (1 or 0) indicating whether the user wishes to generate orthogonalized JLL-based graphs. Default is 0.

`WishForwardPremia`
: A binary variable (1 or 0) indicating whether the user wishes to generate forward premia graphs. Default is 0.

`LimFP`
: A numeric vector containing the maturities associated with the start and end dates of the loan.

`WishBootstrap`
: A binary variable (1 or 0) indicating whether the user wishes to perform bootstrap-based estimation. Default is 0.

`ListBoot`
: A List containing the following four elements:

  1. `methodBS`: Desired bootstrap method: (a) 'bs' for standard residual bootstrap, (b) 'wild' for wild bootstrap, or (c) 'block' for block bootstrap.
  2. `BlockLength`: If block bootstrap is chosen, specify the block length (numeric scalar).
  3. `ndraws`: Number of bootstrap draws.
  4. `pctg`: Confidence level expressed in basis points (numeric vector).

`WishForecast`
: A binary variable (1 or 0) indicating whether the user wishes to generate forecasts. Default is 0.

`ListForecast`
: A list containing the following three elements:

  1. `ForHoriz`: forecast horizon;
  2. `t0Sample`: Index of the first variable in the information set.
  3. `t0Forecast`: Index of the first forecast cut-off date.
  4. `ForType`: A string specifying the desired forecast type. Available options are: "Rolling" or "Expanding".

`UnitYields`
: A character string indicating the maturity unit of yields. Options are: (i) "Month" for yields expressed in months, or (ii) "Year" for yields expressed in years. Default is "Month".

Value

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

Examples

```
ModelType <- "JPS original"
Horiz <- 100
DesiredOutputGraphs <- c("Fit", "GIRF", "GFEVD")
OutputLabel <- "Test"
WishStationarityQ <- 1
WishGraphRiskFac <- 0
WishGraphYields <- 1

InputsList <- InputsForOutputs(ModelType, Horiz, DesiredOutputGraphs, OutputLabel,
                               WishStationarityQ, WishGraphYields, WishGraphRiskFac)
```

IRFandGIRFgraphs *IRF and GIRF graphs for all models*

Description

IRF and GIRF graphs for all models

Usage

```
IRFandGIRFgraphs(
  ModelType,
  NumOut,
  WishPdynamicsgraphs,
  WishYieldsgraphs,
  IRFhoriz,
  PathsGraphs,
  OutputType,
  Economies,
  Folder2save,
  verbose
)
```

Arguments

ModelType	A character vector indicating the estimated model type
NumOut	list of computed outputs containing the model fit, IRFs, FEVDs, GIRFs, GFEVDs and perm premia
WishPdynamicsgraphs	binary variable: set 1, if the user wishes graphs to be generated; or set 0, otherwise

WishYieldsgraphs	binary variable: set 1, if the user wishes graphs to be generated; or set 0, otherwise
IRFhoriz	single numerical vector containing the desired horizon of analysis for the IRFs
PathsGraphs	Path of the folder in which the graphs will be saved
OutputType	Available options are 'IRF' and 'GIRF'
Economies	A character vector containing the names of the economies included in the system.
Folder2save	Folder path where the outputs will be stored.
verbose	Logical flag controlling function messaging.

Available Methods

- 'autoplot(object, type = "IRF_Factor")', 'autoplot(object, type = "IRF_Yields")', 'autoplot(object, type = "GIRF_Yields")', 'autoplot(object, type = "GIRF_Yields")'.

- For JLL-based models other option available are: 'autoplot(object, type = "IRF_Factor-_Ortho")', 'autoplot(object, type = "IRF_Yields_Ortho")', 'autoplot(object, type = "GIRF_Yields_Ortho")', 'autoplot(object, type = "GIRF_Yields_Ortho")'.

Examples

```
data("NumOutEx")
ModelType <- "JPS original"
Economy <- "Brazil"
IRFhoriz <- 20
IRFandGIRFgraphs(ModelType, NumOutEx, WishPdynamicsgraphs = 0, WishYieldsgraphs = 1, IRFhoriz,
                PathsGraphs = NULL, OutputType = "GIRF", Economy, Folder2save = NULL,
                verbose = FALSE)
```

JLL *Estimates the P-dynamics from JLL-based models*

Description

Estimates the P-dynamics from JLL-based models

Usage

```
JLL(NonOrthoFactors, N, JLLinputs, CheckInputs = FALSE)
```

Arguments

`NonOrthoFactors`
: A numeric matrix (F x T) representing the time series of risk factors before the orthogonalization process.

`N`
: Integer. Number of country-specific spanned factors.

`JLLinputs`
: List of necessary inputs to estimate JLL models:

1. Economies: set of economies that are part of the economic system (string-vector)
2. `DomUnit`: A string specifying the name of the economy assigned as the dominant unit.
   If no dominant unit is assigned, set this variable to "None".
3. `WishSigmas`: Set to "1" if the user wishes to estimate the variance-covariance matrices and Cholesky factorizations
   (this can take a long time). Set to "0" if not.
4. `SigmaNonOrtho`: A NULL value or an F x F matrix from the non-orthogonalized dynamics.
5. `JLLModelType`: A string specifying the type of JLL model. Available options are: "JLL original", "JLL joint Sigma", or "JLL No DomUnit".

`CheckInputs`
: A logical flag to indicate whether to perform a prior consistency check on the inputs provided in `JLLinputs`. The default is set to FALSE

Value

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

References

Jotiskhatira, Le and Lundblad (2015). "Why do interest rates in different currencies co-move?" (Journal of Financial Economics)

Examples

```
data(CM_Factors)
RF_TS <- RiskFactors
N <- 3

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

JLLPara <- JLL(RF_TS, N, JLLinputs)
```

LabFac *Generates the labels factors*

Description

Generates the labels factors

Usage

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

Arguments

N	Integer. Number of country-specific spanned factors.
DomVar	A character vector containing the names of the domestic variables.
GlobalVar	A character vector containing the names of the global variables.
Economies	A character vector containing the names of the economies included in the system.
ModelType	A character vector indicating the model type to be estimated.

Value

List containing the risk factor labels

Examples

```
N <- 2
DomVar <- c("inflation", "Output gap")
GlobalVar <- "Commodity Prices"
Economies <- c("U.S.", "Canada", "Germany", "Japan")
ModelType <- "JPS original"

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

LoadData *Loads data sets from several papers*

Description

Loads data sets from several papers

Usage

```
LoadData(DataPaper)
```

Arguments

DataPaper	Available options are BR_2017 (Bauer and Rudebusch, 2017) , CM_2023 (Candelon and Moura, 2023), CM_2024 (Candelon and Moura, 2024)

Value

Complete set of data from several papers.

References

1. Bauer and Rudebusch (2017). "Resolving the Spanning Puzzle in Macro-Finance Term Structure Models" (Review of Finance)
2. Candelon and Moura (2023). "Sovereign yield curves and the COVID-19 in emerging markets" (Economic Modelling)
3. Candelon and Moura (2024). "A Multicountry Model of the Term Structures of Interest Rates with a GVAR" (Journal of Financial Econometrics)

Examples

```
#Example 1:
LoadData("BR_2017")

#Example 2:
LoadData("CM_2023")

#Example 3:
LoadData("CM_2024")
```

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)

ModelParaEx *Example of parameter set after optimization*

Description

JPS-based model for Brazilian data

Usage

```
data("ParaSetEx")
```

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>

NumOutEx — *Example of computed numerical outputs*

Description

Numerical outputs for JPS-based model using Brazilian data

Usage

```
data("NumOutEx")
```

Format

list of inputs and outputs

PC var explained variance explained per spanned factor

Fit model fit of bond yields

IRF Impulse response function

FEVD Forecast error variance decomposition

GIRF Generalized impulse response function

GFEVD Generalized Forecast error variance decomposition

TermPremiaDecomp Term Premia decomposition

NumOutputs — *Constructs the model numerical outputs (model fit, IRFs, GIRFs, FEVDs, GFEVDs, and term premia decomposition)*

Description

Constructs the model numerical outputs (model fit, IRFs, GIRFs, FEVDs, GFEVDs, and term premia decomposition)

Usage

```
NumOutputs(
  ModelType,
  ModelPara,
  InputsForOutputs,
  FactorLabels,
  Economies,
  Folder2save = NULL,
  verbose = TRUE
)
```

Arguments

ModelType	A character vector indicating the model type to be estimated.
ModelPara	A list containing the point estimates of the model parameters. For details, refer to the outputs from the Optimization function.
InputsForOutputs	A list containing the necessary inputs for generating IRFs, GIRFs, FEVDs, GFEVDs and Term Premia.
FactorLabels	A list of character vectors with labels for all variables in the model.
Economies	A character vector containing the names of the economies included in the system.
Folder2save	Folder path where the outputs will be stored. Default option saves the outputs in a temporary directory.
verbose	Logical flag controlling function messaging. Default is TRUE.

Details

Both IRFs and FEVDs are computed using the Cholesky decomposition method. The risk factors are ordered as follows: (i) global unspanned factors, and (ii) domestic unspanned and spanned factors for each country. The order of countries follows the sequence defined in the Economies vector.

Value

An object of class 'ATSMNumOutputs' containing the following keys elements:

1. Model parameter estimates
2. Model fit of bond yields
3. IRFs
4. FEVDs
5. GIRFs
6. GFEVDs
7. Bond yield decomposition

References

Pesaran, H. Hashem, and Shin, Yongcheol. "Generalized impulse response analysis in linear multivariate models." Economics letters 58.1 (1998): 17-29.

Examples

```
data("ParaSetEx")
data("InpForOutEx")
# Adjust inputs according to the loaded features
ModelType <- "JPS original"
Economy <- "Brazil"
FacLab <- LabFac(N = 1, DomVar ="Eco_Act" , GlobalVar = "Gl_Eco_Act", Economy, ModelType)
```

```
NumOut <- NumOutputs(ModelType, ModelParaEx, InpForOutEx, FacLab, Economy,
                     Folder2save = NULL, verbose = FALSE)
```

`Optimization` *Perform the optimization of the log-likelihood function of the chosen ATSM*

Description

Perform the optimization of the log-likelihood function of the chosen ATSM

Usage

```
Optimization(
  MLEinputs,
  StatQ,
  DataFreq,
  FactorLabels,
  Economies,
  ModelType,
  tol = 1e-04,
  TimeCount = TRUE,
  BS_outputs = FALSE,
  verbose = TRUE
)
```

Arguments

`MLEinputs`	A list containing the necessary inputs for building the log-likelihood function (see `InputsForOpt` function).
`StatQ`	A binary variable (1 or 0) indicating whether to impose that the largest eigenvalue under Q is strictly smaller than 1. Set to 1 to impose the restriction, or 0 otherwise.
`DataFreq`	A character vector specifying the data frequency. Available options: "Daily All Days", "Daily Business Days", "Weekly", "Monthly", "Quarterly", "Annually".
`FactorLabels`	A list of character vectors with labels for all variables in the model.
`Economies`	A character vector containing the names of the economies included in the system.
`ModelType`	A character vector indicating the model type to be estimated.
`tol`	Convergence tolerance (scalar). The default is 1e-4.
`TimeCount`	Logical. If TRUE, computes the time required for model estimation. Default is TRUE.
`BS_outputs`	Logical. If TRUE, generates a simplified output list in the bootstrap setting. Default is FALSE.
`verbose`	Logical flag controlling function messaging. Default is TRUE.

Value

An object of class 'ATSMModelOutputs' containing model outputs after the optimization of the chosen ATSM specification.

Available Methods

- 'summary(object)'

References

This function is partially adapted from the `LS__opt` 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

```
LoadData("CM_2024")
ModelType <- "JPS original"
Economy <- "Brazil"
t0 <- "01-05-2007" # Initial Sample Date (Format: "dd-mm-yyyy")
tF <- "01-12-2018" # Final Sample Date (Format: "dd-mm-yyyy")
N <- 1
GlobalVar <- "Gl_Eco_Act" # Global Variables
DomVar <- "Eco_Act" # Domestic Variables
DataFreq <- "Monthly"
StatQ <- 0

FacLab <- LabFac(N, DomVar, GlobalVar, Economy, ModelType)
ATSMInputs <- InputsForOpt(t0, tF, ModelType, Yields, GlobalMacroVar, DomesticMacroVar,
                           FacLab, Economy, DataFreq, CheckInputs = FALSE)

OptPara <- Optimization(ATSMInputs, StatQ, DataFreq, FacLab, Economy, ModelType)
```

Out *Complete list of several outputs from an ATSM*

Description

Example for illustration used in the package vignette

Usage

```
data("Out_Example")
```

Format

several model classes

ModelParaList List of parameter estimates of the selected ATSM

ATSMinputs General inputs from an ATSM

Forecasts List of forecast outputs

NumOut List of numerical outputs

Bootstrap List of set of outputs after bootstrap

---

pca_weights_one_country

Computes the PCA weights for a single country

Description

Computes the PCA weights for a single country

Usage

```
pca_weights_one_country(Yields, Economy)
```

Arguments

Yields	A matrix of bond yields (J x T) for a single country, where J is the number of maturities and T is the time series length.
Economy	A character string indicating the name of the economy.

Value

A matrix (J x J) that corresponds to the eigenvectors of the variance-covariance matrix of yields

Examples

```
data(CM_Yields)
Economy <- "Mexico"
pca_weights <- pca_weights_one_country(Yields, Economy)
```

plot.ATSMModelForecast *Plot method for ATSMModelForecast objects*

Description

Plot method for ATSMModelForecast objects

Usage

```
## S3 method for class 'ATSMModelForecast'
plot(x, ...)
```

Arguments

x	An object of class ATSMModelForecast
...	Additional arguments (not used)

print.ATSMModelInputs *Print method for ATSMModelInputs objects*

Description

Print method for ATSMModelInputs objects

Usage

```
## S3 method for class 'ATSMModelInputs'
print(x, ...)
```

Arguments

x	An object of class 'ATSMModelInputs'
...	Additional arguments (not used)

RiskFactors *Data: Risk Factors - Candelon and Moura (2024, JFEC)*

Description

Risk factors data used in Candelon and Moura (2024, JFEC)

Usage

```
data("CM_Factors")
```

Format

matrix containing the risk factors of the models

References

Candelon, B. and Moura, R. (2024) "A Multicountry Model of the Term Structures of Interest Rates with a GVAR". (Journal of Financial Econometrics)

RiskFactorsGraphs *Spanned and unspanned factors plot*

Description

Spanned and unspanned factors plot

Usage

```
RiskFactorsGraphs(
  ModelType,
  ModelOutputs,
  Economies,
  FactorLabels,
  Folder2save
)
```

Arguments

ModelType	A character vector indicating the estimated model type
ModelOutputs	list of model parameter estimates (see the Optimization function)
Economies	A character vector containing the names of the economies included in the system.
FactorLabels	A character vector indicating the model type to be estimated.
Folder2save	Folder path where the outputs will be stored.

Available Methods

- ‘autoplot(object, type = "RiskFactors")‘

Examples

```
data("ParaSetEx")
# Adapt factor labels according to the example
ModelType <- "JPS original"
Economy <- "Brazil"
FacLab <- LabFac(N = 1, DomVar ="Eco_Act" , GlobalVar = "Gl_Eco_Act", Economy, ModelType)

RiskFactorsGraphs(ModelType, ModelParaEx, Economy, FacLab, Folder2save = NULL)
```

Spanned_Factors *Computes the country-specific spanned factors*

Description

Computes the country-specific spanned factors

Usage

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

Arguments

Yields	A matrix (J x T), where J is the number of maturities and T is the length of the time series.
Economies	A character vector containing the names of the economies included in the system.
N	Scalar representing the desired number of country-specific spanned factors (maximum allowed is N = J).

Value

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

Examples

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

Value

List containg 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,
         dimnames = list(Economies, Economies))
SF <- StarFactors(RiskFactors, Economies, Wgvar)
```

summary.ATSMModelInputs *Summary method for ATSMModelInputs objects*

Description

Summary method for ATSMModelInputs objects

Usage

```
## S3 method for class 'ATSMModelInputs'
summary(object, ...)
```

Arguments

object	An object of class 'ATSMModelInputs'
...	Additional arguments (not used)

---

summary.ATSMModelOutputs *Summary method for ATSMModelOutputs objects*

Description

Summary method for ATSMModelOutputs objects

Usage

```
## S3 method for class 'ATSMModelOutputs'
summary(object, ...)
```

Arguments

object	An object of class 'ATSMModelOutputs'
...	Additional arguments (not used)

TPDecompGraph *Term Premia decomposition graphs for all models*

Description

Term Premia decomposition graphs for all models

Usage

```
TPDecompGraph(
  ModelType,
  NumOut,
  ModelPara,
  WishRPgraphs,
  UnitYields,
  Economies,
  PathsGraphs,
  Folder2Save,
  verbose
)
```

Arguments

ModelType	A character vector indicating the estimated model type
NumOut	list of computed outputs containing the model fit, IRFs, FEVDs, GIRFs, GFEVDs and Risk premia
ModelPara	list of model parameter estimates (See the Optimization function)
WishRPgraphs	binary variable: set 1, if the user wishes graphs to be generated; or set 0, otherwise
UnitYields	(i) "Month": if maturity of yields are expressed in months or (ii) "Year": if maturity of yields are expressed in years
Economies	A character vector containing the names of the economies included in the system.
PathsGraphs	Path of the folder in which the graphs will be saved
Folder2Save	Folder path where the outputs will be stored.
verbose	description Logical flag controlling function messaging.

Available Methods

- ‘autoplot(object, type = "TermPremia")‘

Examples

```
data("ParaSetEx")
data("NumOutEx")
ModelType <- "JPS original"
Economy <- "Brazil"
UnitYields <- "Month"

TPDecompGraph(ModelType, NumOutEx, ModelParaEx, WishRPgraphs = 1, UnitYields, Economy,
              PathsGraphs = NULL, Folder2Save = NULL, verbose = FALSE)
```

TradeFlows *Data: Trade Flows - Candelon and Moura (2024, JFEC)*

Description

Trade Flows data used in Candelon and Moura (2024, JFEC)

Usage

```
data("CM_Trade")
```

Format

list containing the bilateral trade flows

References

Candelon, B. and Moura, R. (2024) "A Multicountry Model of the Term Structures of Interest Rates with a GVAR". (Journal of Financial Econometrics)

`Trade_Flows` *Data: Trade Flows - Candelon and Moura (2023)*

Description

Trade Flows data used in Candelon and Moura (2023)

Usage

```
data("CM_Trade")
```

Format

list containing the bilateral trade flows

References

Candelon, B. and Moura, R. (2023) "Sovereign yield curves and the COVID-19 in emerging markets". (Economic Modelling)

`Transition_Matrix` *Computes the transition matrix required in the estimation of the GVAR model*

Description

Computes the transition matrix required in the estimation of the GVAR model

Usage

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

Arguments

t_First	Sample starting date (in the format: yyyy).
t_Last	Sample ending date (in the format: yyyy).
Economies	A character vector containing the names of the economies included in the system.
type	A character string indicating the method for computing interdependence. Possible options include: • Time-varying: Computes time-varying interdependence and returns the weight matrices for each year based on available data (may extrapolate the sample period). • Sample Mean: Returns a single weight matrix containing the average weights over the entire sample period, suitable for time-invariant interdependence. • A specific year (e.g., "1998", "2005"): Used to compute time-invariant interdependence for the specified year.
DataConnectedness	Data used to compute the transition matrix. Default is set to NULL.
DataPath	Path to the Excel file containing the data (if applicable). The default is linked to the Excel file available in the package.

Details

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"
W_mat <- Transition_Matrix(t_First, t_Last, Economies, type, DataConnectedness = TradeFlows)
```

VAR *Estimates a standard VAR(1)*

Description

Estimates a standard VAR(1)

Usage

```
VAR(RiskFactors, VARtype, Bcon_Mat = NULL)
```

Arguments

RiskFactors	A numeric matrix (F x T) representing the time series of risk factors.
VARtype	String vector with two possible values: 'unconstrained' or 'constrained'.
Bcon_Mat	Constraints matrix (F+1 x N), which includes an intercept. Entries containing NAs are treated as free parameters. Default is set to NULL.

Value

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

Examples

```
data("CM_Factors")
# Example 1: unconstrained case
VAR(RiskFactors, VARtype= 'unconstrained')

# Example 2: constrained case
K <- nrow(RiskFactors)
Bcon_Mat <- matrix(0, nrow = K, ncol = K+1)
Bcon_Mat[ , 1:3] <- NaN
VAR(RiskFactors, VARtype= 'constrained', Bcon_Mat)
```

Yields *Data: Yields - Candelon and Moura (2024, JFEC)*

Description

Yields data used in Candelon and Moura (2024, JFEC)

Bond yield data used in Candelon and Moura (2023)

Usage

```
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. (Forthcoming) "A Multicountry Model of the Term Structures of Interest Rates with a GVAR". (Journal of Financial Econometrics)

Candelon, B. and Moura, R. (2023) "Sovereign yield curves and the COVID-19 in emerging markets". (Economic Modelling)

Index

∗ **(2014)**
BR_jps_out, 8
ModelPara, 29
∗ **-**
InpForOutEx, 20
ModelParaEx, 30
NumOutEx, 31
∗ **Brazil**
InpForOutEx, 20
ModelParaEx, 30
NumOutEx, 31
∗ **Domestic**
DomesticMacroVar, 12
∗ **Factors**
DomMacro, 12
FactorsGVAR, 13
GlobalMacro, 17
GlobalMacroVar, 18
RiskFactors, 37
∗ **Flows**
Trade_Flows, 42
TradeFlows, 41
∗ **GVAR**
DomMacro, 12
FactorsGVAR, 13
∗ **JPS**
BR_jps_out, 8
InpForOutEx, 20
ModelPara, 29
ModelParaEx, 30
NumOutEx, 31
∗ **Numerical**
NumOutEx, 31
∗ **Risk**
DomMacro, 12
FactorsGVAR, 13
GlobalMacro, 17
GlobalMacroVar, 18
RiskFactors, 37
∗ **Trade**
Trade_Flows, 42
TradeFlows, 41
∗ **Yields**
Yields, 44
∗ **examples**
Out, 34
∗ **factors**
DomesticMacroVar, 12
∗ **for**
NumOutEx, 31
∗ **model**
InpForOutEx, 20
ModelParaEx, 30
NumOutEx, 31
∗ **outputs**
BR_jps_out, 8
ModelPara, 29
NumOutEx, 31
∗ **risk**
DomesticMacroVar, 12
∗ **unspanned**
DomesticMacroVar, 12

autoplot, 3
autoplot.ATSMModelBoot, 3
autoplot.ATSMNumOutputs, 4

Bias_Correc_VAR, 4, *7*, *16, 17*
Bootstrap, 6
BR_jps_out, 8

DatabasePrep, 8
DataForEstimation, 10
DomesticMacroVar, 12
DomMacro, 12

FactorsGVAR, 13
FEVDandGFEVDgraphs, 13
Fitgraphs, 15

ForecastYields, 16

GlobalMacro, 17
GlobalMacroVar, 18
GVAR, *5*, *7*, *16*, 18

InpForOutEx, 20
InputsForOpt, *11*, 20, *33*
InputsForOutputs, 23
IRFandGIRFgraphs, 25

JLL, *5*, *7*, *16*, 26

LabFac, 28
LoadData, 28

ModelPara, 29
ModelParaEx, 30
MultiATSM, 30
MultiATSM-package *(*MultiATSM*)*, 30

NumOutEx, 31
NumOutputs, *6*, 31

Optimization, *6*, *15, 16*, *32*, 33, *37*, *41*
Out, 34

pca_weights_one_country, 35
plot.ATSMModelForecast, 36
print.ATSMModelInputs, 36

RiskFactors, 37
RiskFactorsGraphs, 37

Spanned_Factors, 38
StarFactors, 39
summary.ATSMModelInputs, 39
summary.ATSMModelOutputs, 40

TPDecompGraph, 40
Trade_Flows, 42
TradeFlows, 41
Transition_Matrix, *19*, 42

VAR, 43

Yields, 44