

# R documentation

of all in ‘man’

November 15, 2016

## R topics documented:

arimamodelset-class . . . . .	2
arimamodelsetsub-class . . . . .	2
arimamodelsetsub_GetNextModel . . . . .	3
arimamodelsetsub_initialize . . . . .	3
arimamodelsetsub_movetonext . . . . .	3
arimamodelset_initialize . . . . .	4
ldt . . . . .	4
ldt-class . . . . .	4
ldt_initialize . . . . .	5
ldt_Run . . . . .	5
ldt_show . . . . .	6
modelset-class . . . . .	6
modelsetsub-class . . . . .	6
modelsetsub_Run . . . . .	7
modelset_consider . . . . .	7
modelset_considernew . . . . .	7
modelset_GetCounts . . . . .	8
modelset_GetNextModel . . . . .	8
modelset_initialize . . . . .	8
modelset_setSupportedScoringRules . . . . .	9
scoringrule-class . . . . .	9
scoringrule_initialize . . . . .	9
simulationdata-class . . . . .	10
svarxmodelset-class . . . . .	10
svarxmodelsetsub-class . . . . .	11
svarxmodelsetsub_GetNextModel . . . . .	11
svarxmodelsetsub_initialize . . . . .	11
svarxmodelsetsub_movetonext . . . . .	12
svarxmodelset_initialize . . . . .	12

<b>Index</b>	<b>13</b>
--------------	-----------

---

arimamodelset-class	<i>A modelset for ARIMA models</i>
---------------------	------------------------------------

---

### Description

This is a reference class and defines a set of ARIMA (autoregressive integrated moving average) models based on stats package. It contains modelset.

### Fields

**TargetData** A ts that contains the target variable's data. It is set using ldt\$EndoData's first column.

**IsSeasonal** Determines whether the frequency of TargetData is higher than 1.

**XReg** Exogenous variables' data. It is set from ldt\$ExoData. Note that ARIMA models are univariate and therefore in this setting the other endogenous variables in ldt\$EndoData are ignored. If possible, you can use their lagged values as exogenous.

**NewXReg** Data for exogenous variables (i.e., XReg) in the forecast period. Of course, these data are not used in the out-of-sample simulations and therefore model verification; However, currently we restrict our attention to the practical models (i.e., those with which we can forecast the unknown future).

**SimulationData** A list of simulationdata objects. The length of the list is determined by ldt\$SimulationCount.

---

arimamodelsetsub-class	<i>A subset of a arimamodelset</i>
------------------------	------------------------------------

---

### Description

A set of ARIMA models defined in arimamodelset class. It contains modelsetsub.

### Fields

**p** p in ARIMA(p,d,q)x(P,D,Q) model.

**d** d in ARIMA(p,d,q)x(P,D,Q) model.

**q** q in ARIMA(p,d,q)x(P,D,Q) model.

**P** P in ARIMA(p,d,q)x(P,D,Q) model.

**D** D in ARIMA(p,d,q)x(P,D,Q) model.

**Q** Q in ARIMA(p,d,q)x(P,D,Q) model.

**Intercept** Determines whether the ARIMA models in this subset contains intercept.

**ExoIndexes** Determines different exogenous variables based on an indexing approach. A vector of (for example) [2 3] means that the second and third variables in arimamodelset\$XReg are the exogenous variables. This vector will grow in arimamodelsetsub\$GetNextModel.

---

```
arimamodelsetsub_GetNextModel
    see modelsetsub$GetNextModel description.
```

---

**Description**

see modelsetsub\$GetNextModel description.

**Value**

see modelsetsub\$GetNextModel description.

**Fields**

isfirst see modelsetsub\$GetNextModel description.

---

```
arimamodelsetsub_initialize
    The constructor of arimamodelsetsub class
```

---

**Description**

It will generate the required fields in this arimamodelsetsub and its parent modelsetsub.

**Fields**

parentarima The corresponding arimamodelset class.  
 p sets p field of the class.  
 d sets d field of the class.  
 q sets q field of the class.  
 intercept sets Intercept field of the class.  
 P sets P field of the class.  
 D sets D field of the class.  
 Q sets Q field of the class.

---

```
arimamodelsetsub_movetonext
    Changes ExoIndexes field to reach the next ARIMA model
```

---

**Value**

FALSE if no more move is possible. True, if it moved to the next model.

---

```
arimamodelset_initialize
```

*The constructor of arimamodelset class*

---

### Description

It will generate the required fields in this arimamodelset and its parent modelset.

### Fields

parentldt The corresponding ldt class.

---

```
ldt
```

*ldt: A package for probabilistic forecasting*

---

### Description

ldt tries to summarize and compare the results of all other packages that provides a means of time-series forecasting, in a probabilistic approach

### Details

If you have existing Rd files, check out the Rd2roxygen package for a convenient way of converting Rd files to roxygen comments.

### Author(s)

Ramin Mojab <ron3403272@gmail.com>

---

```
ldt-class
```

*Main Class of the Package*

---

### Description

This is a reference class. Generate an instance with `$new` method, change values if needed, and execute `ldt_Run` method to start the estimation, forecast, and evaluation process.

### Fields

EndoData A 'ts' that contains the data for the endogenous variables. The target variable (the one we are trying to forecast) is the first variable in this series.

ExoData A 'matrix' that contains the data for the exogenous variable. note that the frequency and start date will be fetched from EndoData.

MaxHorizon A 'positive integer'. Forecasts will be provided up to this number. Default is 1.

MaxLag A 'positive integer' that indicates the maximum lag of models (including AR and MA parts). Default is 1.

**MaxSize** A 'positive integer' that indicates the maximum size of multiple/multivariate models. Default is 1. For example, in arima case, a value of 2 means that models might have zero or one exogenous variables.

**ScoringRules** A list of 'scoringrule' objects. Default is a list of AIC, BIC, MAE, MSE, LSR, LnSR, QSR, HSR and CRPSR (for all score rules, normality is assumed).

**SimulationCount** A 'positive integer' that indicates the number of out-of-sample simulations; i.e., how many times we should separate the data into training and evaluation samples and test the forecast accuracy. Default is 1.

**ModelSets** A list of 'modelset' objects

### Examples

```
##load("data/endodata_rand.rda")
##load("data/exodata_rand.rda")
##a=ldt$new(endodata,exodata, maxsize = 3)
##a$Run()
```

---

ldt_initialize	<i>Constructor for ldt Class</i>
----------------	----------------------------------

---

### Description

initialize ldt reference class by using ldt\$new(...).

### Arguments

endodata	sets EndoData field of the class.
exodata	sets ExoData field of the class.
maxhorizon	sets MaxHorizon field of the class.
maxlag	sets MaxLag field of the class.
maxsize	sets MaxSize field of the class.
simulationcount	sets SimulationCount field of the class.

---

ldt_Run	<i>Starts the Forecasting Process</i>
---------	---------------------------------------

---

### Description

Use this function to start estimation/forecast/evaluation process.

### Note

I separate the run block from initialize, because it is reference class and it is important to give the user a chance to change the fields. That's why any validation must be provided here.

ldt_show	<i>Summerizes this class</i>
modelset-class	<i>Represents a specific model set</i>

### Description

A model set is a collection of different statistical models that share one or more properties. For example, AR(p) for  $p=1\dots 5$  can be considered as a set of autoregressive models.

### Fields

ParentLDT The main ldt class

ID A unique string that represents the related package

Description A description for the related package

SupportedScoringRules A list of scoringrules that the models in this modelset support. E.g., not all models can report AIC.

Subs A list of modelsetsub.

Results A list that contains the results for each member of SupportedScoringRules. Results[[i]][[h]] represents the best model with respect to the i-th rule in SupportedScoringRules and for horizon h.

AllModels In some cases you might want to save all the models.

modelsetsub-class	<i>A subset of a modelset</i>
-------------------	-------------------------------

### Description

It is sometimes practical to partition a modelset into some other subsets.

### Fields

ParentSet The corresponding modelset.

CountRequired A integer that represents the size of this subset.

CountValid A integer that represents the current number of successful estimations/forecasts. By successful I mean no singular matrix inversion or no other types of errors.

CountFailed A integer that represents the current number of failed estimations/forecasts. Some of these failures are due to the provided data (user) and some are due to the implemented package.

---

modelsetsub_Run	<i>Starts the forecasting process in this modelsetsub</i>
-----------------	---

---

### Description

This method provides a loop and estimates one model after the other.

### Value

None

---

modelset_consider	<i>Considers a new estimated model</i>
-------------------	--

---

### Details

This method converts the forecast errors and the forecast standard errors to a score, based on scoringrule\$Score field. Then, it sends the provided scores to modelset\$considernew for comparison purposes.

### Value

FALSE if modelerrorse is empty (a failure), TRUE, otherwise.

### Fields

modelerrorse A vector with three elements: [[1]] The estimated model, [[2]] The forecast error, [[3]] The forecast standard error. If model has failed to forecast, it should be an empty vector (i.e., length(modelerrorse) == 0).

---

modelset_considernew	<i>Considers a new estimated model</i>
----------------------	--

---

### Description

Each model must generate a score for each forecast horizons. This method will compare the generated scores with the best ones and determines whether to discard this model or keep it.

### Arguments

model	The model based on which the scores are generated.
i	The index of scoringrule in SupportedScoringRules field, based on which the scores are generated.
scores	The generated scores as a vector. i-th element is for the i-th horizon.

---

modelset_GetCounts	<i>Reports the current state of the process</i>
--------------------	---

---

### Description

It reports the number of required, valid and invalid estimations based on information provided in the Subs field.

### Value

an array with three elements: [[1]] CountRrequired, [[2]] CountValid, [[3]] CountFailed

---

modelset_GetNextModel	<i>Handles moving from one model to the next</i>
-----------------------	--

---

### Description

(You should overwrite this method) Models will be estimated and evaluated one after the other. This method should provide the required algorithm.

### Value

NULL if there is no more move, An empty list if any error occurred, A list of 1. Model 2. Forecast error and 3. forecast standard error otherwise.

### Fields

isfirst Determines whether it is the first model in this modelsetsub.

---

modelset_initialize	<i>The constructor of modelset class</i>
---------------------	--

---

### Details

The Results field is prepared in this constructor. Set the SupportedScoringRules field (and other fields) before executing callSuper() command in the inherited classes.



---

modelset_setSupportedScoringRules	<i>Sets SupportedScoringRules field</i>
-----------------------------------	---

---

**Description**

Sets SupportedScoringRules field based on a given list of IDs (strings). Note that a scoringrule with such an ID must be presented in ParentLDT\$ScoringRules, or an error will be raised.

**Value**

None

**Fields**

namelist A list of strings that each represent an ID of a scoringrule in ParentLDT\$ScoringRules.

---

scoringrule-class	<i>Represents a scoring rule</i>
-------------------	----------------------------------

---

**Description**

A scoringrule measures the accuracy of probabilistic forecasts in an out-of-sample simulation practice.

**Fields**

ID A short string that describes this scoringrule.

Description A string that provides other information about this scoringrule.

IsPositivelyOriented A boolean indicates whether higher values are better.

Score The scoring function. It should get two arguments: vector of forecast errors and vector of forecast standard errors and it should return a vector of the same length

---

scoringrule_initialize	<i>The constructor of scoringrule class</i>
------------------------	---

---

**Arguments**

id	sets ID field of the class
description	sets Description field of the class
ispositiveoriented	sets Ispositiveoriented field of the class
score	sets Score field of the class

---

simulationdata-class    *Required data to simulate a forecast*

---

### Description

It provides the required data to forecast and calculate the forecast errors and forecast standard errors in a simulation practice.

### Fields

TrainingSampleTarget    A ts for the target variable that will be used in estimation.

ValidationSample    A matrix for the target variables that will be used in validation.

TrainingSampleOther    A matrix that contains other information for estimation of the model.

NewXRegValidation    A matrix of exogenous variables in the validation sample.

---

svarxmodelset-class    *A modelset for Stationary VARX models*

---

### Description

This is a reference class and defines a set of Stationary VAR (vector autoregressive) models with stationary exogenous variables, based on stats package. It contains modelset.

### Fields

TargetData    A ts that contains the target variable's data. It is set using ldt\$EndoData's first column.

EndoExoData    A matrix that contains other types of data; i.e., other endogenous and exogenous ones. Endogenous data comes first.

ExoStartIndex    The index of the column in EndoExoData which contains the first exogenous variable.

NewExoData    A matrix that contains the future values of the exogenous variables.

SimulationData    A list of simulationdata objects. The length of the list is determined by ldt\$SimulationCount.

---

svarxmodelsetsub-class

*A subset of a svarxmodelset*

---

### Description

A set of Stationary VARX models defined in svarxmodelset class. It contains modelsetsub.

### Fields

P The number of lags of the model.

Size The number of different endogenous and exogenous variables in the model.

Intercept Determines whether this model has an intercept.

Indexes P Determines different combinations of endogenous and exogenous variables. A vector of (for example) [2 3] means that the second and third variables in svarxmodelset\$EndoExoData are in this model. This vector will change in svarxmodelsetsub\$GetNextModel.

---

svarxmodelsetsub\_GetNextModel

*see modelsetsub\_GetNextModel description.*

---

### Description

see modelsetsub\_GetNextModel description.

### Value

see modelsetsub\_GetNextModel description.

### Fields

isfirst see modelsetsub\_GetNextModel description.

---

svarxmodelsetsub\_initialize

*The constructor of svarxmodelsetsub class*

---

### Details

It will generate the required fields in this svarxmodelsetsub and its parent modelsetsub.

### Fields

parentsvarx The corresponding arimamodelset class.

p sets p field of the class.

size sets size field of the class.

intercept sets Intercept field of the class.

---

`svarxmodelsetsub_movetonext`*Changes Indexes field to reach the next St. VARX model*

---

**Value**

FALSE if no more move is possible. True, if it moved to the next model.

---

`svarxmodelset_initialize`*The constructor of svarxmodelset class*

---

**Description**

It will generate the required fields in this svarxmodelset and its parent modelset.

**Fields**

parentldt The corresponding ldt class.

# Index

arimamodelset (arimamodelset-class), 2  
arimamodelset-class, 2  
arimamodelset\_initialize, 4  
arimamodelsetsub  
    (arimamodelsetsub-class), 2  
arimamodelsetsub-class, 2  
arimamodelsetsub\_GetNextModel, 3  
arimamodelsetsub\_initialize, 3  
arimamodelsetsub\_movetonext, 3  
  
ldt, 4  
ldt (ldt-class), 4  
ldt-class, 4  
ldt-package (ldt), 4  
ldt\_initialize, 5  
ldt\_Run, 4, 5  
ldt\_show, 6  
  
modelset (modelset-class), 6  
modelset-class, 6  
modelset\_consider, 7  
modelset\_considernew, 7  
modelset\_GetCounts, 8  
modelset\_GetNextModel, 8  
modelset\_initialize, 8  
modelset\_setSupportedScoringRules, 9  
modelsetsub (modelsetsub-class), 6  
modelsetsub-class, 6  
modelsetsub\_Run, 7  
  
scoringrule (scoringrule-class), 9  
scoringrule-class, 9  
scoringrule\_initialize, 9  
simulationdata (simulationdata-class),  
    10  
simulationdata-class, 10  
svarxmodelset (svarxmodelset-class), 10  
svarxmodelset-class, 10  
svarxmodelset\_initialize, 12  
svarxmodelsetsub  
    (svarxmodelsetsub-class), 11  
svarxmodelsetsub-class, 11  
svarxmodelsetsub\_GetNextModel, 11  
svarxmodelsetsub\_initialize, 11  
svarxmodelsetsub\_movetonext, 12