Package ‘multisensi’

May 20, 2010

Type Package

Title Multivariate Sensitivity Analysis

Version 1.0-3

Date 2010-05-20

Author Matieyendou LAMBONI <clamboni@yahoo.fr>, Herve MONOD <herve.monod@jouy.inra.fr>

Maintainer Herve MONOD <herve.monod@jouy.inra.fr>

Description An R library for performing sensitivity analysis on a model with multivariate output

Repository CRAN

LazyLoad yes

Depends R (>= 2.8.0)

Suggests MASS

R topics documented:

`multisensi-package` *Multivariate sensitivity Analysis*

Description

Sensitivity Analysis (SA) for models with multivariate output

Details

This package contains three methods for performing sensitivity analysis on simulation models with multivariate output:

i) `gsi` function for the Generalised Sensitivity Analysis (Lamboni et al., 2009) based on inertia decomposition. This method synthesizes the information that is spread between the time outputs or between the principal components and produces a unique sensitivity index for each factor.

ii) `gsi` function for the componentwise sensitivity analysis obtained by computing sensitivity indices on principal components (Campbell et al., 2006)

iii) `dynsi` function for the dynamic sensitivity analysis obtained by computing sensitivity indices on each output variable.

For all three methods, sensitivity indices are calculated presently by using a factorial design and a classical ANOVA decomposition.

Simulation model management

The multisensi package works on simulation models coded either in R or using an external language (typically as an executable file). Models coded in R must be either functions or objects that have a predict method, such as lm objects. Models defined as functions will be called once with an expression of the form `y <- f(X)` where `X` is a vector containing a combination of levels of the input factors, and `y` is the output vector of length q, where q is the number of output variables. If the model is external to R, for instance a computational code, it must be analyzed with the decoupled approach: the methods require an input data frame (`X`) containing all the combinations of the input levels and the outputs data frame (`Y`) containing the response of the model corresponding to these combinations. The size of `X` is $n * p$ and the size of `Y` is $n * q$ where p is the number of the input factor, q is the number of the model outputs and n is the number of all the combinations of the input levels. This approach can also be used on R models that do not fit the required specifications.

Author(s)

Matieyendou Lamboni <clamboni@yahoo.fr>, Herv\'e Monod <herve.monod@jouy.inra.fr>

References

Lamboni, M., Makowski, D., Monod, H., 2009. Multivariate global sensitivity analysis for dynamic crop models. Field Crops Research, volume 113. pp. 312-320

Lamboni, M., Makowski, D., Monod, H., 2009. Multivariate sensitivity analysis to measure global contribution of input factors in dynamic models (submitted in october 2009 to Reliability Engineering \& System Safety

Saltelli, A., Chan, K., Scott, E.M. eds, 2000. *Sensitivity Analysis* Wiley, New York.

`acp` *Principal Component Analysis (PCA)*

Description

Principal Component Analysis (PCA) for the generalized sensitivity analysis

Usage

```
ACP(simuls, normalized = TRUE)
```

Arguments

`simuls`	data frame, typically a multivariate model output
`normalized`	If TRUE, a normalized PCA is performed

Value

an object of class prcomp; see the `prcomp` help for further details

Note

This is essentially an internal function for the multisensi package

Author(s)

M. LAMBONI

See Also

prcomp

anoasg — *Dynamic main and total sensitivity indices*

Description

Function to compute the main and total dynamic sensitivity indices

Usage

```
anoasg(ANO, nbcomp = 2)
```

Arguments

`ANO`	anova object obtained from the anovadec function
`nbcomp`	number of model output to be considered

Value

A list with the following components:

SI data frame of first order, second order, ... indices

tSI data frame of total sensitivity

mSI data frame of main sensitivity indices

iSI data frame of interaction sensitivity indices

indic.fact data frame of anova object attribute

Note

This is essentially an internal function for the multisensi package

Author(s)

M. LAMBONI

anovadec — *Computation of several anovas on the output of a PCA*

Description

A function to compute sum of squares decomposition on principal components by using the aov function

Usage

```
anovadec(ACP, plan, ord.inter, nbcomp = 2)
```

Arguments

`ACP`	data.frame of model output, usually the x component of a PCA object
`plan`	data.frame of input design
`ord.inter`	ANOVA formula like `"A+B+c+A:B"` OR an integer giving the maximum interaction order (1 for main effects)
`nbcomp`	number of principal components to be considered (e.g 3)

Value

The anovadec function returns a two-component list:

aov list of AOV objects

PC prediction of output

Note

This is essentially an internal function for the multisensi package

Author(s)

M. LAMBONI

See Also

aov

`asg` *Main and total generalized sensitivity computation*

Description

Function to compute the main and total generalized sensitivity indices

Usage

```
asg(ANO, ACP, sigma.car, nbcomp = 2)
```

Arguments

`ANO`	ANOVA Object obtained from anovadec function
`ACP`	ACP object
`sigma.car`	Inertia for the model output
`nbcomp`	Numbers of principal component (PC) to be considered

Value

A list with the following components:

SI data frame of first order, second order, ... indices

mSI data frame of main sensitivity indices

tSI data frame of total sensitivity

iSI data frame of interaction sensitivity indices

cor data frame of correlation between PCs and model output

inertia Vector of Inertia explained by PCs

indic.fact data frame of anova object attribute

Note

This is essentially an internal function for the multisensi package

Author(s)

M. LAMBONI

`biomasse` *The Winter Wheat Dynamic Model*

Description

The Winter Wheat Dynamic Model, a toy model to illustrate the main multisensi methods

Usage

```
biomasse(input, climdata, annee = 3)
```

Arguments

`input`	vector of input value
`annee`	year
`climdata`	a meteorological data.frame specific to biomasse

Details

The Winter Wheat Dry Matter model (WWDM) is a dynamic crop model running at a daily time step (Makowski et al, 2004). It has two state variables, the above-ground winter wheat dry matter $U(t)$, in g/m^2 and the leaf area index LAI(t) with t the day number from sowing ($t = 1$) to harvest ($t = 223$). In the multisensi-package implementation, the `biomasse` function simulates the output for only one parameter set (the first row of `input` if it is a matrix or a data.frame).

Value

a vector of daily dry matter increase of the Winter Wheat biomass, over 223 days

Author(s)

initially Makowski, D., 2004

References

Makowski, D., Jeuffroy, M.-H., Gu\'erif, M., 2004 Bayesian methods for updating crop model predictions, applications for predicting biomass and grain protein content. In: Bayesian Statistics and Quality Modelling in the Agro-Food Production Chain (van Boeakel et al. eds), pp. 57-68. Kluwer, Dordrecht

Monod, H., Naud, C., Makowski, D., 2006 Uncertainty and sensitivity analysis for crop models. In: Working with Dynamic Crop Models (Wallach D., Makowski D. and Jones J. eds), pp. 55-100. Elsevier, Amsterdam

---

`Climat` *Climate data*

Description

Climate data for the WWDM model (needed by the `biomasse` function)

Usage

```
data(Climat)
```

Format

A data frame with 3126 observations on the following 4 variables.

`ANNEE` a factor with levels 1 to 14, indicating 14 different years

`RG` daily radiation variable

`Tmin` daily maximum temperature

`Tmax` daily minimum temperature

Source

Makowski, D., Jeuffroy, M.-H., Gu\'erif, M., 2004 Bayesian methods for updating crop model predictions, applications for predicting biomass and grain protein content. In: Bayesian Statistics and Quality Modelling in the Agro-Food Production Chain (van Boeakel et al. eds), pp. 57-68. Kluwer, Dordrecht.

Monod, H., Naud, C., Makowski, D., 2006 Uncertainty and sensitivity analysis for crop models. In: Working with Dynamic Crop Models (Wallach D., Makowski D. and Jones J. eds), pp. 55-100. Elsevier, Amsterdam `dynsi` *Dynamic Sensitivity Indices: DSI*

Description

dynsi implements the Dynamic Sensitivity Indices. This method allows to compute classical Sensitivity Indices on each output variable of a dynamic or multivariate model by using the ANOVA decomposition

Usage

```
dynsi(formula, model, factors, cumul = FALSE, simulonly=FALSE,
      nb.outp = NULL, Name.File=NULL, ...)
```

Arguments

`formula`	ANOVA formula like `"A+B+c+A:B"` OR an integer equal to the maximum interaction order in the sensitivity model
`model`	output data.frame OR the name of the R-function which calculates the model output. The only argument of this function must be a vector containing the input factors values
`factors`	input data.frame (the design) if model is a data.frame OR a list of factors levels such as `factor.example <- list(A=c(0,1),B=c(0,1,4))`
`cumul`	logical value. If TRUE the sensitivity analysis will be done on the cumalative outputs
`simulonly`	logical value. If TRUE the program stops after calculating the design and the model outputs
`nb.outp`	The first nb.outp number of model outputs to be considered. If NULL all the outputs are considered
`Name.File`	optional name of a R script file containing the R-function that calculates the simulation model. e.g `"exc.ssc"`
`...`	possible fixed parameters of the model function

Details

If factors is a list of factors, the dynsi function generates a complete factorial design. If it is a data.frame, dynsi expects that each column is associated with an input factor.

Value

dynsi returns a list of class "dynsi", containing all the input arguments detailed before, plus the following components:

`X`	a data.frame containing the experimental design (input samples)
`Y`	a data.frame containing the output matrix (response)
`SI`	a data.frame containing the Sensitivity Indices (SI) on each output variable of the model
`mSI`	a data.frame of principal SI on each output variable

`tSI`	a data.frame containing the total SI on each output variable
`iSI`	a data.frame of interaction SI on each output variable

...

Author(s)

Matieyendou LAMBONI

References

M. Lamboni, D. Makowski and H. Monod, 2009. Multivariate global sensitivity analysis for dynamic crop models. Field Crops Research, 113, 312-320.

A. Saltelli, K. Chan and E. M. Scott eds, 2000. *Sensitivity Analysis*. Wiley, New York.

See Also

gsi

Examples

```
##---- Should be DIRECTLY executable !! ----
# Test case : the Winter Wheat Dynamic Models (WWDM)
# input factors design,
 data(plan)
# input Climate variables
 data(Climat)
 DYNSI <- dynsi(2, biomasse, plan, nb.outp=10, climdata=Climat)
 summary(DYNSI)
 print(DYNSI)
 plot(DYNSI)
 #graph.bar(DYNSI,col=1, beside=F)  # sensitivity bar plot
                                    # for the first output (col=1)
 #graph.bar(DYNSI,col=2, xmax=1)    #
```

`graph.bar` *Sensitivity index bar plot*

Description

A function that plots sensitivity indices by a bar graph

Usage

```
graph.bar(x, col = 1, nb.plot = 15, xmax = NULL, beside = TRUE, ...)
```

Arguments

`x`	an object of class gsi or dynsi
`col`	the column number of GSI to represent in the bar graph
`nb.plot`	number of input factors to be considered
`xmax`	a user-defined maximal x value ($x \leq 1$) in all the bar graphs that show sensitivity indices; or NULL if the user wants to keep default values
`beside`	if TRUE, the main and total sensitivity indices are represented by two bars; if FALSE, they are represented by the same bar
`...`	graphical parameters

Author(s)

M. LAMBONI

`graph.pc` *Principal Components graph for gsi objects*

Description

A function that plots the Principal components (PCs) and the sensitivity index on each PC

Usage

```
graph.pc(x, nb.plot = 15, nb.comp = NULL, xmax = NULL,
         beside = TRUE, ...)
```

Arguments

`x`	gsi object
`nb.plot`	number of input factors to be considered
`nb.comp`	number of PCs
`xmax`	a user-defined maximal x value ($x \leq 1$) in all the bar graphs that show sensitivity indices; or NULL if the user wants to keep default values
`beside`	if TRUE, the main and total sensitivity indices are represented by two bars; if FALSE, they are represented by the same bar
`...`	graphical parameters

Author(s)

M. LAMBONI

`grpe.gsi` *Group factor GSI, obsolete function*

Description

An obsolete function that computed the GSI of a group factor as one factor

Usage

```
grpe.gsi(GSI, fact.interet)
```

Arguments

`GSI`	a gsi or dynsi object
`fact.interet`	input factor to be grouped

Note

This is essentially an internal function for the multisensi package

Author(s)

M. LAMBONI

`gsi` *Generalized Sensitivity Indices: GSI*

Description

The gsi function implements the calculation of Generalised Sensitivity Indices. This method allows to compute a synthetic Sensitivity Index for the dynamic or multivariate models by using factorial designs and the MANOVA decomposition of inertia. It computes also the Sensitivity Indices on principal components

Usage

```
gsi(formula, model, factors, inertia = 0.95, normalized = TRUE,
    cumul = FALSE, simulonly = FALSE, Name.File = NULL, ...)
```

Arguments

`formula`	ANOVA formula like `"A+B+C+A:B"` OR an integer equal to the maximum interaction order in the sensitivity model
`model`	output data.frame OR the name of the R-function which calculates the model output. The only argument of this function must be a vector containing the input factors values
`factors`	input data.frame (the design) if model is a data.frame OR a list of factors levels such as : `factor.example <- list(A=c(0,1),B=c(0,1,4))`

`inertia`	cumulated proportion of inertia (a scalar < 1) to be explained by the selected Principal components OR number of PCs to be used (e.g 3)
`normalized`	logical value. TRUE (default) computes a normalized Principal Component analysis.
`cumul`	logical value. If TRUE the PCA will be done on the cumulative outputs
`simulonly`	logical value. If TRUE the program stops after calculating the design and the model outputs
`Name.File`	optional name of a R script file containing the R-function that calculates the simulation model. e.g `"exc.ssc"`
`...`	possible fixed parameters of the model function

Details

If factors is a list of factors, the gsi function generates a complete factorial design. If it is a data.frame, gsi expects that each column is associated with an input factor.

Value

gsi returns a list of class "gsi", containing all the input arguments detailed before, plus the following components:

`X`	a data.frame containing the experimental design (input samples)
`Y`	a data.frame containing the output matrix (response)
`SI`	a data.frame containing the Sensitivity Indices (SI) on PCs and the Generalized SI (GSI)
`mSI`	a data.frame of first order SI on PCs and first order GSI
`tSI`	a data.frame containing the total SI on PCs and the total GSI
`iSI`	a data.frame of interaction SI on PCs and interaction GSI
`cor`	a data.frame of correlation between PCs and outputs
`inertia`	vector of inertia per PCs and global criterion
`Rsquare`	vector of dynamic coefficient of determination

Author(s)

M. Lamboni

References

M. Lamboni, D. Makowski and H. Monod, 2009. Multivariate global sensitivity analysis for dynamic crop models. Field Crops Research, volume 113. pp. 312-320

M. Lamboni, D. Makowski and H. Monod, 2009. Multivariate sensitivity analysis to measure global contribution of input factors in dynamic models. Submitted to Reliability Engineering and System Safety.

See Also

dynsi

Examples

```
##---- Should be DIRECTLY executable !! ----
# Test case : the Winter Wheat Dynamic Models (WWDM)
#  input factors design
 data(plan)
# input climate variable
 data(Climat)
 GSI <- gsi(2, biomasse, plan, inertia=3, normalized=TRUE, cumul=FALSE, climdata=Climat)
 summary(GSI)
 print(GSI)
 plot(x=GSI, beside=FALSE)
 #plot(GSI, nb.plot=4)          # the 'nb.plot' most influent factors
                                # are represented in the plots
 #plot(GSI,nb.comp=2, xmax=1)   # nb.comp = number of principal components
 #plot(GSI,nb.comp=3, graph=1) # graph=1 for first figure; 2 for 2nd one
                                # and 3 for 3rd one; or 1:3 etc.
 #graph.bar(GSI,col=1, beside=F)  # sensitivity bar plot on the first PC
 #graph.bar(GSI,col=2, xmax=1)    #
```

`plan` *A factorial input design for the main example*

Description

Factorial design (resolution V) data for the 7 WWDM model input factors

Usage

```
data(plan)
```

Format

A data frame with 2187 observations on the following 7 variables.

`Eb` First WWDM input factor name

`Eimax` Second WWDM input factor name

`K` Thirth WWDM input factor name

`Lmax` Fourth WWDM input factor name

`A` Fifth WWDM input factor name

`B` Sixth WWDM input factor name

`TI` Seventh WWDM input factor name

See Also

`biomasse`

Examples

```
data(plan)
## maybe str(plan) ; plot(plan) ...
```

planfact — *Complete factorial design in lexical order*

Description

Function that generates a complete factorial design in lexical order

Usage

```
planfact(nb.niv, make.factor = TRUE)
```

Arguments

`nb.niv`	vector containing the number of each input levels
`make.factor`	logical value. If TRUE the columns of the output are of class factor

Value

`plan`	data frame of the complete factorial design

Note

This is essentially an internal function for the multisensi package

Author(s)

M. LAMBONI

planfact.as — *Complete factorial design*

Description

Computation of a complete factorial design for model input factors

Usage

```
planfact.as(input)
```

Arguments

`input`	list of factor levels

Value

`comp2`	complete factorial design of model input

Note

This is essentially an internal function for the multisensi package

Author(s)

M. LAMBONI

`plot.dynsi` *Plot method for dynamic sensitivity results*

Description

Plot method for dynamic sensitivity results of class dynsi

Usage

```
## S3 method for class 'dynsi':
plot(x, text.tuning = NULL, ...)
```

Arguments

`x`	a dynsi object
`text.tuning`	NULL or a small integer to improve the position of input factor labels
`...`	graphical parameters

Details

For labels that would be partly positioned outside the plot frame, the argument "text.tuning" may allow to get a better positioning. If it is equal to n, say, these labels are moved by n positions inside the frame, where 1 position corresponds to 1 output variable on the x-axis.

Author(s)

M. LAMBONI

See Also

`dynsi`

`plot.gsi` *Plot method for generalised sensitivity analysis*

Description

Plot method for generalised sensitivity analysis of class gsi

Usage

```
## S3 method for class 'gsi':
plot(x, nb.plot = 10, nb.comp = 3, graph = 1:3, xmax=NULL,
                    beside=TRUE,...)
```

Arguments

`x`	a gsi object
`nb.plot`	number of input factors to be considered
`nb.comp`	number of Principal Components to be plotted
`graph`	figures number: 1 or 2 or 3
`xmax`	a user-defined maximal x value ($x \leq 1$) in all the bar graphs that show sensitivity indices; or NULL if the user wants to keep default values
`beside`	if TRUE, the main and total sensitivity indices are represented by two bars; if FALSE, they are represented by the same bar
`...`	graphical parameters

Author(s)

M. LAMBONI

See Also

`gsi`

`print.dynsi` *print DYNSI*

Description

A function to print DYNSI results

Usage

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

Arguments

`x`	a dynsi object
`...`	print parameters

Author(s)

M. LAMBONI

See Also

`dynsi`

print.gsi *print GSI*

Description

function to print GSI results

Usage

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

Arguments

`x`	a gsi object
`...`	print parameters

Author(s)

M. LAMBONI

See Also

`gsi`

quality *quality of any approximation*

Description

Function that computes the sensitivity quality after making some assumptions about the number of PCs and the number of interactions

Usage

```
quality(echsimul, echsimul.app, normalise = TRUE)
```

Arguments

`echsimul`	model outputs
`echsimul.app`	Predicted model output
`normalise`	logical value

Value

A list with the following components:

moy mean

biais biais

coef.det R-square

Note

This is essentially an internal function for the multisensi package

Author(s)

M. LAMBONI

`simulmodel` *Model simulation*

Description

Function that simulates the model outputs

Usage

```
simulmodel(model, plan, nomFic = NULL, verbose = FALSE, ...)
```

Arguments

`model`	name of R-function
`plan`	data frame of input design
`nomFic`	name of file that contains the model function
`verbose`	verbose
`...`	... possible fixed parameters of the R-function

Details

The model function must be a R-functions. Models defined as functions will be called once with an expression of the form `y <- f(X)` where `X` is a vector containing a combination of levels of the input factors, and `y` is the output vector of length q, where q is the number of output variables

Value

data frame of model outputs

Note

This is essentially an internal function for the multisensi package

Author(s)

M. LAMBONI

`summary.dynsi` *dynsi summary*

Description

Function to summarize the dynamic sensitivity results

Usage

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

Arguments

`object`	a dynsi object
`...`	summary parameters

Author(s)

M. LAMBONI

See Also

`dynsi`

`summary.gsi` *summary of GSI results*

Description

function to summarize the GSI results

Usage

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

Arguments

`object`	a GSI object
`...`	summary parameters

Author(s)

M. LAMBONI

See Also

`gsi`

yapprox — *Prediction based on PCA and anovas*

Description

A function that predicts the model output after PCA and aov analyses

Usage

```
yapprox(ACP, nbcomp = 2, aov.obj)
```

Arguments

`ACP`	prcomp object
`nbcomp`	number of PCs
`aov.obj`	aov object

Value

A list with components

Y model output predictions

trace model output inertia

Note

This is essentially an internal function for the multisensi package

Author(s)

M. LAMBONI

Index