Package 'sensobol'

September 20, 2019

Title Computation of High-Order Sobol' Sensitivity Indices

Version 0.2.1

Description It allows to rapidly compute, bootstrap and plot up to third-order Sobol' indices using the estimators by Saltelli et al. 2010 <doi:10.1016/j.cpc.2009.09.018> and Jansen 1999 <doi:10.1016/S0010-4655(98)00154-4>. The 'sensobol' package also implements the algorithm by Khorashadi Zadeh et al. 2017 <doi:10.1016/j.envsoft.2017.02.001> to calculate the approximation error in the computation of Sobol' first and total indices, an approach that allows to robustly screen influential from non-influential model inputs. Finally, it also provides functions to obtain publication-ready figures of the model output uncertainty and sensitivity-related analysis.

License GPL-3

Encoding UTF-8

Imports boot (>= 1.3.20), data.table (>= 1.12.0), ggplot2 (>= 3.1.0), magrittr (>= 1.5), matrixStats (>= 0.54.0), randtoolbox (>= 1.17.1), Rdpack (>= 0.7), rlang (>= 0.3.1), scales (>= 1.0.0), stats, stringr (>= 1.4.0), utils

RdMacros Rdpack

LazyData true

Depends R (>= 3.4.0)

RoxygenNote 6.1.1

Suggests knitr, rmarkdown, testthat, covr

VignetteBuilder knitr

URL http://github.com/arnaldpuy/sensobol

BugReports http://github.com/arnaldpuy/sensobol/issues

NeedsCompilation no

Author Arnald Puy [aut, cre] (<https://orcid.org/0000-0001-9469-2156>), Bertrand Ioos [ctb] (Author of included 'sensitivity' fragments), Gilles Pujol [ctb] (Author of included 'sensitivity' fragments), RStudio [cph] (Copyright holder of included 'sensitivity' fragments)

Maintainer Arnald Puy <arnald.puy@gmail.com>

Repository CRAN

Date/Publication 2019-09-20 13:50:02 UTC

R topics documented:

`bratley1988_Fun` *Bratley and Fox (1988) function*

Description

It implements the Bratley and Fox (1988) function, which requires *n* model inputs.

Usage

```
bratley1988_Fun(X)
```

Arguments

`X`	A data frame or numeric matrix.

Value

A numeric vector with the model output.

Examples

```
A <- sobol_matrices(n = 100, k = 6)
Y <- bratley1988_Fun(A)
```

bratley1992_Fun *Bratley, Fox and Niederreiter (1992) function*

Description

It implements the Bratley et al. (1992) function, #' which requires *n* model inputs.

Usage

```
bratley1992_Fun(X)
```

Arguments

X	A data frame or numeric matrix.

Value

A numeric vector with the model output.

References

Bratley P, Fox BL, Niederreiter H (1992). "Implementation and tests of low-discrepancy sequences." *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, **2**(3), 195–213.

Examples

```
A <- sobol_matrices(n = 100, k = 4)
Y <- bratley1992_Fun(A)
```

ishigami_Mapply *Ishigami function*

Description

It implements the Ishigami and Homma (1990) function, which requires 3 model inputs. The transformation of the distribution of the model inputs (from U(0, 1) to U(-pi, +pi)) is conducted internally.

Usage

```
ishigami_Mapply(X)
```

Arguments

X	A data frame or numeric matrix.

Value

A numeric vector with the model output.

References

Ishigami T, Homma T (1990). "An importance quantification technique in uncertainty analysis for computer models." *Proceedings. First International Symposium on Uncertainty Modeling and Analysis*, **12**, 398–403.

Examples

```
A <- sobol_matrices(n = 100, k = 3)
Y <- ishigami_Mapply(A)
```

---

oakley_Fun *Oakley & O'Hagan (2004) function*

Description

It implements the Oakley and O'Hagan (2004) function. The function needs 15 model inputs. The transformation of the distribution of the model inputs (from uniform to normally distributed with mean = 0 and sd = 1) is conducted internally.

Usage

```
oakley_Fun(X)
```

Arguments

X	A data frame or numeric matrix.

Value

A numeric vector with the model output.

References

Oakley J, O'Hagan A (2004). "Probabilistic sensitivity analysis of complex models: a Bayesian approach." *Journal of the Royal Statistical Society B*, **66**(3), 751–769. ISSN 13697412, doi: 10.1111/j.14679868.2004.05304.x.

Examples

```
A <- sobol_matrices(n = 100, k = 15)
Y <- oakley_Fun(A)
```

plot_scatter *Scatterplots of the model output against the model inputs*

Description

Scatterplots of the model output against the model inputs

Usage

```
plot_scatter(x, Y, n, params)
```

Arguments

x	A data table, data frame or matrix with the model inputs.
Y	Numeric vector with the model output.
n	Integer, sample size of the Sobol' matrix.
params	Vector with the name of the model inputs.

Value

A ggplot object.

Examples

```
# Define settings:
n <- 100; k <- 8; R <- 10
# Design the sample matrix:
A <- sobol_matrices(n = n, k = k, second = TRUE, third = TRUE)
# Compute the model output:
Y <- sobol_Fun(A)
# Plot scatterplots:
plot_scatter(x = A, Y = Y, n = n, params = colnames(data.frame(A)))
```

plot_sobol *Plot Sobol' first and total-order indices*

Description

Plot Sobol' first and total-order indices

Usage

```
plot_sobol(x, dummy = NULL, type = 1)
```

Arguments

x	A data.table.
dummy	The output of the sobol_ci_dummy function. If supplied and type = 1, the plot includes an horizontal transparent frame showing the confidence intervals of the first and total-order indices for the dummy parameter.
type	An integer. If type = 1, it plots first and total effects. If type = 2, it plots second-order effects. If type = 3, it plots third-order effects. Default is type = 1.

Value

A ggplot object.

Examples

```
# Define settings:
n <- 500; k <- 8; R <- 100
# Design the sample matrix:
A <- sobol_matrices(n = n, k = k, second = TRUE, third = TRUE)
# Compute the model output:
Y <- sobol_Fun(A)
# Compute the Sobol' indices:
sens <- sobol_indices(Y = Y, params = colnames(data.frame(A)),
R = R, n = n, parallel = "no", ncpus = 1, second = TRUE, third = TRUE)
# Compute the Sobol' indices for the dummy parameter:
s.dummy <- sobol_dummy(Y = Y, params = colnames(data.frame(A)), R = R, n = n)
# Compute confidence intervals:
sens.ci <- sobol_ci(sens, params = colnames(data.frame(A)), type = "norm", conf = 0.95)
# Compute confidence intervals for the dummy parameter:
s.dummy.ci <- sobol_ci_dummy(s.dummy, type = "norm", conf = 0.95)
# Plot Sobol' indices:
plot_sobol(sens.ci, dummy = s.dummy.ci, type = 1)
```

---

plot_uncertainty *Plot model output uncertainty*

Description

It creates an histogram with the model output distribution.

Usage

```
plot_uncertainty(Y, n = NULL)
```

Arguments

Y	A numeric vector with the model output.
n	An integer with the initial sample size.

Value

a ggplot2 object.

Examples

```
# Define settings:
n <- 100; k <- 8; R <- 10
# Design the sample matrix:
A <- sobol_matrices(n = n, k = k, second = FALSE, third = FALSE)
# Compute the model output:
Y <- sobol_Fun(A)
# Plot the model output distribution:
plot_uncertainty(Y, n = n)
```

---

sobol_ci *Bootstrap confidence intervals for Sobol' indices.*

Description

It computes bootstrap confidence intervals for Sobol' indices.

Usage

```
sobol_ci(b, params, type, conf, second = FALSE, third = FALSE)
```

Arguments

b	The output of the `sobol_indices` function.
params	A vector with the name of the model inputs.
type	A vector of character strings representing the type of intervals required. The value should be any subset of the values `c("norm","basic","perc","bca")`. For more information, check the function `boot.ci`.
conf	A scalar or vector containing the confidence level(s) of the required interval(s).
second	Logical. If `second = TRUE`, it computes the confidence intervals for second-order indices. Default is `second = FALSE`.
third	Logical. If `third = TRUE`, it computes the confidence intervals for third-order indices. Default is `third = FALSE`.

Value

A data table.

See Also

`boot`, `boot.ci`.

Examples

```
# Define settings:
n <- 1000; k <- 8; R <- 100
# Design the sample matrix:
A <- sobol_matrices(n = n, k = k, second = TRUE, third = TRUE)
# Compute the model output:
Y <- sobol_Fun(A)
# Compute the Sobol' indices:
sens <- sobol_indices(Y = Y, params = colnames(data.frame(A)),
R = R, n = n, parallel = "no", ncpus = 1,
second = TRUE, third = TRUE)
# Compute confidence intervals:
sobol_ci(sens, params = colnames(data.frame(A)), type = "norm", conf = 0.95)
```

---

sobol_ci_dummy *Bootstrap confidence intervals for the dummy parameter*

Description

It computes bootstrap confidence intervals for the dummy parameter.

Usage

```
sobol_ci_dummy(b, type = type, conf = conf)
```

Arguments

b	The output of the sobol_dummy function.
type	A vector of character strings representing the type of intervals required. The value should be any subset of the values c("norm","basic","perc","bca"). For more information, check the function boot.ci.
conf	A scalar or vector containing the confidence level(s) of the required interval(s).

Value

A data table.

See Also

boot, boot.ci.

Examples

```
# Define settings:
n <- 100; k <- 8; R <- 10
# Design the sample matrix:
A <- sobol_matrices(n = n, k = k, second = TRUE, third = TRUE)
# Compute the model output:
Y <- sobol_Fun(A)
# Compute the Sobol' indices for the dummy parameter:
s.dummy <- sobol_dummy(Y = Y, params = colnames(data.frame(A)), R = R, n = n)
# Compute the confidence intervals for the dummy parameter:
sobol_ci_dummy(s.dummy, type = "norm", conf = 0.95)
```

`sobol_dummy` *Computation of Sobol' indices for a dummy parameter*

Description

This function computes first and total-order Sobol' indices for a dummy parameter following the formulas shown in Khorashadi Zadeh et al. (2017).

Usage

```
sobol_dummy(Y, params, R, n, parallel = "no", ncpus = 1)
```

Arguments

`Y`	Numeric vector, model output.
`params`	Vector with the name of the model inputs.
`R`	Integer, number of bootstrap replicas.
`n`	Integer, sample size of the sample matrix.
`parallel`	The type of parallel operation to be used (if any). If missing, the default is taken from the option "boot.parallel" (and if that is not set, "no"). For more information, check the `parallel` option in the `boot` function of the `boot` package.
`ncpus`	Integer: number of processes to be used in parallel operation: typically one would chose this to the number of available CPUs. Check the `ncpus` option in the `boot` function of the `boot` package.

Value

A data.table object. It includes a column with the results of the bootstrap.

References

Khorashadi Zadeh F, Nossent J, Sarrazin F, Pianosi F, van Griensven A, Wagener T, Bauwens W (2017). "Comparison of variance-based and moment-independent global sensitivity analysis approaches by application to the SWAT model." *Environmental Modelling and Software*, **91**, 210–222. ISSN 13648152, doi: 10.1016/j.envsoft.2017.02.001, http://dx.doi.org/10.1016/j.envsoft.2017.02.001.

See Also

Check the function boot for further details on the bootstrapping and the components available within the class boot.

Examples

```
# Define settings:
n <- 100; k <- 8; R <- 10
# Design the sample matrix:
A <- sobol_matrices(n = n, k = k, second = TRUE, third = TRUE)
# Compute the model output:
Y <- sobol_Fun(A)
# Compute the Sobol' indices for the dummy parameter:
sobol_dummy(Y = Y, params = colnames(data.frame(A)), R = R, n = n)
```

---

sobol_Fun *Sobol' G function*

Description

It implements the Sobol' (1998) function. In this case, the function works with 8 model inputs.

Usage

```
sobol_Fun(X)
```

Arguments

X	A data frame or numeric matrix.

Value

A numeric vector with the model output.

References

Sobol' IM (1998). "On quasi-Monte Carlo integrations." *Mathematics and Computers in Simulation*, **47**(2-5), 103–112. ISSN 03784754, doi: 10.1016/S03784754(98)000962, http://linkinghub.elsevier.com/retrieve/pii/S0378475498000962.

Examples

```
A <- sobol_matrices(n = 100, k = 8)
Y <- sobol_Fun(A)
```

`sobol_indices` *Computation of first, second, third and total-order Sobol' indices*

Description

It computes and bootstraps up to third-order Sobol' indices using either the Saltelli et al. (2010) or the Jansen (1999) estimator.

Usage

```
sobol_indices(Y, params, type = "jansen", R, n, parallel = "no",
  ncpus = 1, second = FALSE, third = FALSE)
```

Arguments

`Y`	Numeric vector, model output.
`params`	Vector with the name of the model inputs.
`type`	Estimator to use: `type = "saltelli"` uses the Saltelli et al. (2010) estimator; `type = "jansen"` uses the Jansen (1999) estimator. Default is `type = "jansen"`.
`R`	Integer, number of bootstrap replicas.
`n`	Integer, sample size of the sample matrix.
`parallel`	The type of parallel operation to be used (if any). If missing, the default is taken from the option "boot.parallel" (and if that is not set, "no"). For more information, check the `parallel` option in the `boot` function of the `boot` package.
`ncpus`	Integer: number of processes to be used in parallel operation: typically one would chose this to the number of available CPUs. Check the `ncpus` option in the `boot` function of the `boot` package.
`second`	Logical. if `second = TRUE`, it computes second-order Sobol' indices.
`third`	Logical. if `third = TRUE`, it computes third-order Sobol' indices.

Value

A data.table object. It includes a column with the results of the bootstrap.

References

Jansen M (1999). "Analysis of variance designs for model output." *Computer Physics Communications*, **117**(1), 35–43. ISSN 00104655, doi: 10.1016/S00104655(98)001544.

Saltelli A, Annoni P, Azzini I, Campolongo F, Ratto M, Tarantola S (2010). "Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index." *Computer Physics Communications*, **181**(2), 259–270. ISSN 00104655, doi: 10.1016/j.cpc.2009.09.018.

See Also

Check the function boot for further details on the bootstrapping and the components available within the class boot.

Examples

```
# Define settings:
n <- 1000; k <- 8; R <- 100
# Design the sample matrix:
A <- sobol_matrices(n = n, k = k, second = TRUE, third = TRUE)
# Compute the model output:
Y <- sobol_Fun(A)
# Compute the Sobol' indices:
sens <- sobol_indices(Y = Y, params = colnames(data.frame(A)),
R = R, n = n, parallel = "no", ncpus = 1, second = TRUE, third = TRUE)
```

---

sobol_matrices *Creation of the sample matrices*

Description

It creates the sample matrices to compute Sobol' first and total-order indices. If needed, it also creates the sample matrices required to compute second and third-order indices. It uses Sobol' quasi-random number sequences.

Usage

```
sobol_matrices(n, k, second = FALSE, third = FALSE, cluster = NULL)
```

Arguments

n	Integer, sample size of the Sobol' matrix.
k	Integer, number of model inputs.
second	Logical. If second = TRUE, it creates the scrambled matrix required to compute second-order indices. Default is second = FALSE.
third	Logical. If third = TRUE, it creates the scrambled matrix required to compute third-order indices. Default is third = FALSE.
cluster	List of vectors, each vector including the column number of the parameters forming the cluster. The default is cluster = NULL

Details

If cluster = NULL, the function generates an $(n, 2k)$ matrix using Sobol' quasi-random number sequences. The first k-matrix is the **A** matrix and the remaining k-matrix, the **B** matrix. It then generates k additional matrices (**A^j_B**), $j = 1, 2, ..., k$, where the k matrix is composed of all columns of the **A** matrix except the j-th column, which is the j column of the **B** matrix. This

approach leds to a total number of model runs of $n(k+2)$ for first and total-order indices (Saltelli et al. 2010).

If a list of vectors is assigned to `cluster`, the output is a scrambled matrix with all columns from matrix **A** except those of the parameters included in the cluster, which come from matrix **B**.

Value

A matrix.

References

Saltelli A, Annoni P, Azzini I, Campolongo F, Ratto M, Tarantola S (2010). "Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index." *Computer Physics Communications*, **181**(2), 259–270. ISSN 00104655, doi: 10.1016/j.cpc.2009.09.018.

See Also

Check the function `sobol` in the package `randtoolbox` to see how the Sobol' quasi-random number sequences are constructed.

Examples

```
sobol_matrices(n = 100, k = 8, second = TRUE, third = TRUE)
```

---

sobol_replicas *Extract bootstrap replicas*

Description

It creates a `data.table` with all the bootstrap replicas.

Usage

```
sobol_replicas(dt, k, second = FALSE, third = FALSE)
```

Arguments

`dt`	The output of the `sobol_indices` function.
`k`	The number of parameters.
`second`	Logical. If `second = TRUE`, it computes the confidence intervals for second-order indices. Default is `second = FALSE`.
`third`	Logical. If `third = TRUE`, it computes the confidence intervals for third-order indices. Default is `third = FALSE`.

Value

A `data.table`.

Examples

```
# Define settings:
n <- 100; k <- 8; R <- 100
# Design the sample matrix:
A <- sobol_matrices(n = n, k = k, second = TRUE, third = TRUE)
# Compute the model output:
Y <- sobol_Fun(A)
# Compute the Sobol' indices:
sens <- sobol_indices(Y = Y, params = colnames(data.frame(A)),
R = R, n = n, parallel = "no", ncpus = 1, second = TRUE, third = TRUE)
# Extract the bootstrap replicas up to third-order
replicas <- sobol_replicas(dt = sens, k = k, second = TRUE, third = TRUE)
```

Index