

Package: UncDecomp (via r-universe)

October 4, 2026

Title Uncertainty Decomposition

Version 1.0.1

Author Seonghyeon Kim [aut, cre], Yongdai Kim [aut], Ilsang Ohn [aut]

Maintainer Seonghyeon Kim <shkim93@snu.ac.kr>

Description If a procedure consists of several stages and there are several models that can be selected for each stage, uncertainty of the procedure can be decomposed by stages or models. This package includes the ANOVA-based method, the cumulative uncertainty-based method, and the balanced decomposition method. Yongdai Kim et al. (2019)
<[doi:10.1016/j.hydroa.2019.100024](https://doi.org/10.1016/j.hydroa.2019.100024)> is a related paper which is accessible via the URL below.

Depends R (>= 3.4.4)

License GPL-2

URL <https://www.sciencedirect.com/science/article/pii/S2589915519300082>

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

NeedsCompilation no

Imports ggplot2, dplyr

Repository <https://kim-seonghyeon.r-universe.dev>

Date/Publication 2019-11-07 10:00:05 UTC

RemoteUrl <https://github.com/cran/UncDecomp>

RemoteRef HEAD

RemoteSha 0133c6ee36ffa41125de6af5f0ef86efdca18f71

Contents

apply_sweep	2
cum_uncertainty	3

drange	4
plotUDlist	5
printUD	6
scenario_uncertainty	7
stage_uncertainty	8
UD_ANOVA_model	9
UD_ANOVA_stage	10
UD_bal_model	11
UD_bal_stage	12
UD_cum_stage	13
UD_model2stage	14
UDtable	15
unc_measures	16
unc_postprocess	17
var0	18

Index	19
--------------	-----------

apply_sweep	<i>Slight modifications of apply() and sweep()</i>
-------------	--

Description

Slightly modified version of apply() and sweep(). apply0() and sweep0() are modification of apply() and sweep() so that they can be used when the length(MARGIN) is zero. msweep() is a modification of sweep() so that it can be used when function receives multiple summary statistic.

Usage

```
apply0(X, MARGIN, FUN, ...)
```

```
sweep0(X, MARGIN, STATS, FUN, ...)
```

```
msweep(X, MARGIN, STATS, FUN, ...)
```

Arguments

X	an array.
MARGIN	apply0() : a vector giving the subscripts which the function will be applied over. sweep0(), msweep() : a vector of indices giving the extent(s) of x which correspond to STATS.
FUN	the function to be applied. For msweep(), a function that receives the elements of X and list in order
...	further arguments passed to or from other methods.
STATS	the summary statistic array which is to be swept out. For msweep(), list of summary statistic array.

Value

If each call to FUN returns a vector of length n, then apply() returns an array of dimension c(n, dim(X)[MARGIN]) if n > 1. If n equals 1, apply() returns a scalar if MARGIN has length 0, a vector if MARGIN has length 1 and an array of dimension dim(X)[MARGIN] otherwise. sweep0() and msweep() return an array with the same shape as x, but with the summary statistics swept out.

Examples

```
set.seed(0)
A <- array(rnorm(24), dim = 4:2)
meanA0 <- apply0(A, numeric(0), mean)
meanA12 <- apply0(A, 1:2, mean)
sdA12 <- apply0(A, 1:2, sd)
ctrArray <- function(a,mu) return(a-mu)
sweep0(A, numeric(0), meanA0, ctrArray)
sweep0(A, 1:2, meanA12, ctrArray)
statsA12 <- list(meanA12, sdA12)
stdArray <- function(a,mu,sigma) return((a-mu)/sigma)
msweep(A, 1:2, statsA12, stdArray)
```

cum_uncertainty

*Cumulative uncertainty(DEPRECATED)***Description**

This function performs uncertainty decomposition based on the cumulative uncertainty.

Usage

```
cum_uncertainty(data, var_name, stages = setdiff(names(data), var_name),
  U = var0)
```

Arguments

data	a data frame containing scenarios(factor or character) for each stages and the variable of interest(numeric). data should contain all combinations of scenarios.
var_name	the name of the variable of interest
stages	names of the stages in the modeling chain. should be ordered by the order of the modeling chain
U	a function that returns uncertainty such as range and variance of a given numeric vector. This package have built-in uncertainty functions var0() and drange(). Default is var0().

Value

summary of uncertainties

Examples

```
stage1 <- LETTERS[1:3]
stage2 <- LETTERS[1:2]
stage3 <- LETTERS[1:4]
y <- rnorm(3*2*4)
data <- expand.grid(stage1=stage1,
                    stage2=stage2,
                    stage3=stage3)
data <- cbind(data, y)
# cum_uncertainty() is deprecated. Use UD_cum_stage()
# cum_uncertainty(data,"y", names(data)[-4])
# cum_uncertainty(data,"y", names(data)[-4],drange)
```

drange

Range(DEPRECATED)

Description

This function returns the difference of maximum and minimum of a given vector.

Usage

```
drange(x)
```

Arguments

x a numeric vector.

Value

the difference of maximum and minimum of a given vector

Examples

```
(x <- rnorm(5))
# drange() is deprecated. Use U_range()
# drange(x)
```

plotUDlist

*Plot UD_list***Description**

This function plots how ppud's result changes as lambda increases.

Arguments

x	output of ppud(UD_list class)
lwd	line width in plot
...	further arguments passed to or from other methods.

Value

ggplot showing how the result of ppud changes as lambda increases

Examples

```

set.seed(0)
stage1 <- LETTERS[1:3]
stage2 <- LETTERS[1:2]
stage3 <- LETTERS[1:4]
y <- rnorm(3*2*4)
data <- expand.grid(stage1=stage1,
                   stage2=stage2,
                   stage3=stage3)
stages <- names(data)
data <- cbind(data, y)

UD_bal_model_range <- UD_bal_model(data, "y", stages, u_range, flist_range)
UD_bal_model_range

UD_bal_stage_range <- UD_model2stage(UD_bal_model_range)
UD_bal_stage_range

UD_model_list = ppud(UD_bal_model_range)
plot(UD_model_list)

UD_stage_list = ppud(UD_bal_stage_range)
plot(UD_stage_list)

```

printUD	<i>Print UD</i>
---------	-----------------

Description

This function creates a table summarizing the uncertainty.

Usage

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

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

Arguments

x	model wise uncertainty(UD_model class) or stage wise uncertainty(UD_stage class)
...	further arguments passed to or from other methods.

Examples

```
set.seed(0)
stage1 <- LETTERS[1:3]
stage2 <- LETTERS[1:2]
stage3 <- LETTERS[1:4]
y <- rnorm(3*2*4)
data <- expand.grid(stage1=stage1,
                   stage2=stage2,
                   stage3=stage3)
stages <- names(data)
data <- cbind(data, y)

UD_bal_model_range <- UD_bal_model(data, "y", stages, u_range, flist_range)
print(UD_bal_model_range)

UD_bal_stage_range <- UD_model2stage(UD_bal_model_range)
print(UD_bal_stage_range)
```

scenario_uncertainty *Scenario uncertainty(DEPRECATED)*

Description

This function performs uncertainty decomposition by scenario based on the second order interaction ANOVA model. The uncertainty from interaction effect from two scenarios is divided equally and assigned to each scenario.

Usage

```
scenario_uncertainty(data, var_name, stages = setdiff(names(data),
  var_name))
```

Arguments

data	a data frame containing scenarios(factor or character) for each stages and the variable of interest(numeric). data should contain all combinations of scenarios.
var_name	the name of the variable of interest
stages	names of the stages of interest.

Value

List of 4 elements

summary	summary of uncertainties
main_uncer	list of which element is a vector of uncertainties from the main effects of scenarios in the corresponding stage
int_uncer	list of which element is a vector of uncertainties from the interaction effects of scenarios in the corresponding stage
scenario_uncer	list of which element is a vector of uncertainties of scenarios in the corresponding stage

Examples

```
stage1 <- LETTERS[1:3]
stage2 <- LETTERS[1:2]
stage3 <- LETTERS[1:4]
y <- rnorm(3*2*4)
data <- expand.grid(stage1=stage1,
  stage2=stage2,
  stage3=stage3)
data <- cbind(data, y)
# scenario_uncertainty() is deprecated. Use UD_ANOVA_model()
# scenario_uncertainty(data,"y", names(data)[-4])
```

stage_uncertainty	<i>Stage uncertainty(DEPRECATED)</i>
-------------------	--------------------------------------

Description

This function performs uncertainty decomposition by stage based on the second order interaction ANOVA model. The uncertainty from interaction effect from two stages is divided equally and assigned to each stage.

Usage

```
stage_uncertainty(data, var_name, stages = setdiff(names(data),
  var_name))
```

Arguments

data	a data frame containing scenarios(factor or character) for each stages and the variable of interest(numeric). data should contain all combinations of scenarios. columns scenarios
var_name	the name of the variable of interest
stages	names of the stages in the model.

Value

List of 4 elements	
summary	summary of uncertainties
main_uncer	a vector of uncertainties from the main effects of the stages
int_uncer	a vector of uncertainties from the interaction effects of the stages
scenario_uncer	a vector of uncertainties of the stages

Examples

```
stage1 <- LETTERS[1:3]
stage2 <- LETTERS[1:2]
stage3 <- LETTERS[1:4]
y <- rnorm(3*2*4)
data <- expand.grid(stage1=stage1,
  stage2=stage2,
  stage3=stage3)
data <- cbind(data, y)
# stage_uncertainty() is deprecated. Use UD_ANOVA_stage()
# stage_uncertainty(data,"y", names(data)[-4])
```

UD_ANOVA_model

*Model-wise uncertainty based on the second order interaction ANOVA***Description**

This function performs uncertainty decomposition by model based on the second order interaction ANOVA model. The uncertainty from interaction effect from two models is divided equally and assigned to each model.

Usage

```
UD_ANOVA_model(data, var_name, stages = setdiff(names(data), var_name))
```

Arguments

data	a data frame containing models(factor or character) for each stages and the variable of interest(numeric). data should contain all combinations of models.
var_name	the name of the variable of interest
stages	names of the stages of interest.

Value

List(UD_model class) including uncertainties of models, uncertainties from main effects, uncertainties from interaction, total uncertainty, names of stages and models

Examples

```
set.seed(0)
stage1 <- LETTERS[1:3]
stage2 <- LETTERS[1:2]
stage3 <- LETTERS[1:4]
y <- rnorm(3*2*4)
data <- expand.grid(stage1=stage1,
                   stage2=stage2,
                   stage3=stage3)
stages <- names(data)
data <- cbind(data, y)

UD_ANOVA_model <- UD_ANOVA_model(data, "y", stages)
UD_ANOVA_model
UD_ANOVA_stage <- UD_model2stage(UD_ANOVA_model)
UD_ANOVA_stage
```

UD_ANOVA_stage	<i>Stage-wise uncertainty based on the second order interaction ANOVA</i>
----------------	---

Description

This function performs uncertainty decomposition by stage based on the second order interaction ANOVA model. The uncertainty from interaction effect from two stages is divided equally and assigned to each stage.

Usage

```
UD_ANOVA_stage(data, var_name, stages = setdiff(names(data), var_name))
```

Arguments

data	a data frame containing models(factor or character) for each stages and the variable of interest(numeric). data should contain all combinations of models.
var_name	the name of the variable of interest
stages	names of the stages in the model.

Value

List(UD_stage class) including uncertainties of stages, uncertainties from main effects, uncertainties from interaction, total uncertainty, names of stages.

Examples

```
set.seed(0)
stage1 <- LETTERS[1:3]
stage2 <- LETTERS[1:2]
stage3 <- LETTERS[1:4]
y <- rnorm(3*2*4)
data <- expand.grid(stage1=stage1,
                   stage2=stage2,
                   stage3=stage3)
stages <- names(data)
data <- cbind(data, y)
UD_ANOVA_stage(data,"y", stages)
```

UD_bal_model	<i>Model-wise balanced uncertainty</i>
--------------	--

Description

This function performs the balanced uncertainty decomposition. In balanced uncertainty decomposition, we assume that the total uncertainty decomposes into the uncertainty of all main effects and all orders of interaction between models. This method distributes the uncertainties of each element evenly among the associated models.

Usage

```
UD_bal_model(data, var_name, stages, u = u_var, flist = flist_var)
```

Arguments

data	a data frame containing models(factor or character) for each stages and the variable of interest(numeric). data should contain all combinations of models.
var_name	the name of the variable of interest
stages	names of the stages of interest.
u	a function that returns uncertainty of each element of the vector like difference or square of difference between each element and summary statistics. This package have built-in uncertainty functions u_var(), u_mad() and u_range(). Default is u_var().
flist	list of functions that summarize vector like mean or median. This package have built-in uncertainty functions flist_var(), flist_mad() and flist_range(). Default is flist_var().

Value

model-wise uncertainties(UD_model class)

Examples

```
set.seed(0)
stage1 <- LETTERS[1:3]
stage2 <- LETTERS[1:2]
stage3 <- LETTERS[1:4]
y <- rnorm(3*2*4)
data <- expand.grid(stage1=stage1,
                   stage2=stage2,
                   stage3=stage3)
stages <- names(data)
data <- cbind(data, y)

UD_bal_model_var <- UD_bal_model(data, "y", stages, u_var, flist_var)
UD_bal_model_var
```

```

UD_bal_model_mad <- UD_bal_model(data, "y", stages, u_mad, flist_mad)
UD_bal_model_mad
UD_bal_model_range <- UD_bal_model(data, "y", stages, u_range, flist_range)
UD_bal_model_range

UD_bal_stage_var <- UD_model2stage(UD_bal_model_var)
UD_bal_stage_var
UD_bal_stage_mad <- UD_model2stage(UD_bal_model_mad)
UD_bal_stage_mad
UD_bal_stage_range <- UD_model2stage(UD_bal_model_range)
UD_bal_stage_range

UD_list = ppud(UD_bal_model_range)
plot(UD_list)

lambda = min_lambda_calc(UD_bal_model_range, 0.01)
UD_1percent = ppud(UD_bal_model_range, lambda)
UD_1percent$UD[[1]]

```

UD_bal_stage	<i>Stage-wise balanced uncertainty</i>
--------------	--

Description

This function performs the balanced uncertainty decomposition. In balanced uncertainty decomposition, we assume that the total uncertainty decomposes into the uncertainty of all main effects and all orders of interaction between stages. This method distributes the uncertainties of each element evenly among the associated stages.

Usage

```
UD_bal_stage(data, var_name, stages, U = U_var)
```

Arguments

data	a data frame containing models(factor or character) for each stages and the variable of interest(numeric). data should contain all combinations of models.
var_name	the name of the variable of interest
stages	names of the stages of interest.
U	a function that returns uncertainty such as range and variance of a given numeric vector. This package have built-in uncertainty functions U_var(), U_mad() and U_range(). Default is U_var().

Value

stage-wise uncertainties(UD_stage class)

Examples

```

set.seed(0)
stage1 <- LETTERS[1:3]
stage2 <- LETTERS[1:2]
stage3 <- LETTERS[1:4]
y <- rnorm(3*2*4)
data <- expand.grid(stage1=stage1,
                   stage2=stage2,
                   stage3=stage3)
stages <- names(data)
data <- cbind(data, y)
UD_bal_stage(data, "y", stages, U_var)
UD_bal_stage(data, "y", stages, U_mad)
UD_bal_stage(data, "y", stages, U_range)

```

UD_cum_stage

*Stage-wise uncertainty based on cumulative uncertainty***Description**

This function performs uncertainty decomposition based on the cumulative uncertainty.

Usage

```
UD_cum_stage(data, var_name, stages = setdiff(names(data), var_name),
            U = U_var)
```

Arguments

data	a data frame containing models(factor or character) for each stages and the variable of interest(numeric). data should contain all combinations of models.
var_name	the name of the variable of interest
stages	names of the stages in the modeling chain. should be ordered by the order of the modeling chain
U	a function that returns uncertainty such as range and variance of a given numeric vector. This package have built-in uncertainty functions U_var(), U_mad() and U_range(). Default is U_var().

Value

stage-wise uncertainties(UD_stage class)

Examples

```

set.seed(0)
stage1 <- LETTERS[1:3]
stage2 <- LETTERS[1:2]
stage3 <- LETTERS[1:4]
y <- rnorm(3*2*4)
data <- expand.grid(stage1=stage1,
                    stage2=stage2,
                    stage3=stage3)
stages <- names(data)
data <- cbind(data, y)

UD_cum_stage(data, "y", stages, U_var)
UD_cum_stage(data, "y", stages, U_mad)
UD_cum_stage(data, "y", stages, U_range)

```

UD_model2stage

*Convert model uncertainty to stage uncertainty***Description**

This function converts model uncertainty to stage uncertainty by summing by stage.

Usage

```
UD_model2stage(UD)
```

Arguments

UD model wise uncertainty(UD_model class), output of function that returns model wise uncertainty such as UD_bal_model and UD_ANOVA_model

Value

stage wise uncertainties(UD_stage class)

Examples

```

set.seed(0)
stage1 <- LETTERS[1:3]
stage2 <- LETTERS[1:2]
stage3 <- LETTERS[1:4]
y <- rnorm(3*2*4)
data <- expand.grid(stage1=stage1,
                    stage2=stage2,
                    stage3=stage3)
stages <- names(data)
data <- cbind(data, y)

```

```

UD_bal_model_var <- UD_bal_model(data, "y", stages, u_var, flist_var)
UD_bal_model_var
UD_bal_model_mad <- UD_bal_model(data, "y", stages, u_mad, flist_mad)
UD_bal_model_mad
UD_bal_model_range <- UD_bal_model(data, "y", stages, u_range, flist_range)
UD_bal_model_range

UD_bal_stage_var <- UD_model2stage(UD_bal_model_var)
UD_bal_stage_var
UD_bal_stage_mad <- UD_model2stage(UD_bal_model_mad)
UD_bal_stage_mad
UD_bal_stage_range <- UD_model2stage(UD_bal_model_range)
UD_bal_stage_range

```

UDtable	<i>Make uncertainty table</i>
---------	-------------------------------

Description

This function summarizes the uncertainty into a table.

Usage

```

UD_table(UD, ...)

## S3 method for class 'UD_model'
UD_table(UD, include.nat = TRUE, include.tot = TRUE,
  ...)

## S3 method for class 'UD_stage'
UD_table(UD, include.nat = TRUE, include.tot = TRUE,
  ...)

```

Arguments

UD	model-wise uncertainty or stage-wise uncertainty
...	further arguments passed to or from other methods. include.nat or include.tot
include.nat	If include.nat is TRUE and UD has nat_unc, create a table containing it.
include.tot	If include.tot is TRUE, create a table containing it.

Value

uncertainty table

Examples

```

set.seed(0)
stage1 <- LETTERS[1:3]
stage2 <- LETTERS[1:2]
stage3 <- LETTERS[1:4]
y <- rnorm(3*2*4)
data <- expand.grid(stage1=stage1,
                   stage2=stage2,
                   stage3=stage3)
stages <- names(data)
data <- cbind(data, y)

UD_bal_model_range <- UD_bal_model(data, "y", stages, u_range, flist_range)
UD_bal_model_range

UD_bal_stage_range <- UD_model2stage(UD_bal_model_range)
UD_bal_stage_range

UD_table(UD_bal_model_range)
UD_table(UD_bal_stage_range)

```

unc_measures

uncertainty measures

Description

Functions beginning with U are uncertainty measure that return a scalar given a vector, such as mean absolute deviation or variance. Functions beginning with flist are lists of functions that summarize vector like mean or median. Functions beginning with u are the uncertainty of each element of the vector, which is calculated from the elements of the vector and summary statistics of the vector, like difference or square of difference between two values.

Usage

```

U_range(x)

u_range(x, a, b, n)

flist_range

U_mad(x)

u_mad(x, m)

flist_mad

U_var(x)

```

```
u_var(x, m)
```

```
flist_var
```

Arguments

x	a vector.
a, b, n	summary statistics for u_range(), a and b are minimum and maximum of x. n is length of x
m	summary statistics. median for u_mad() and mean for u_var()

Format

An object of class list of length 3.

unc_postprocess	<i>Postprocess UD</i>
-----------------	-----------------------

Description

ppud() adjusts uncertainty so that it is not less than a certain value. In particular, it is often used for UD_bal_model, where the uncertainty may be negative. min_lambda_calc() finds lambda such that the proportion of the minimum uncertainty is the specified value.

Usage

```
ppud(UD, lambda_list)
```

```
min_lambda_calc(UD, prop)
```

Arguments

UD	model-wise uncertainty or stage-wise uncertainty
lambda_list	a numeric vector that adjust the degree to which uncertainties and average uncertainty are close.
prop	target proportion of least uncertainty

Examples

```
set.seed(0)
stage1 <- LETTERS[1:3]
stage2 <- LETTERS[1:2]
stage3 <- LETTERS[1:4]
y <- rnorm(3*2*4)
data <- expand.grid(stage1=stage1,
                   stage2=stage2,
                   stage3=stage3)
```

```

stages <- names(data)
data <- cbind(data, y)

UD_bal_model_var <- UD_bal_model(data, "y", stages, u_var, flist_var)
UD_bal_model_var
UD_bal_model_mad <- UD_bal_model(data, "y", stages, u_mad, flist_mad)
UD_bal_model_mad
UD_bal_model_range <- UD_bal_model(data, "y", stages, u_range, flist_range)
UD_bal_model_range

UD_bal_stage_var <- UD_model2stage(UD_bal_model_var)
UD_bal_stage_var
UD_bal_stage_mad <- UD_model2stage(UD_bal_model_mad)
UD_bal_stage_mad
UD_bal_stage_range <- UD_model2stage(UD_bal_model_range)
UD_bal_stage_range

UD_list = ppud(UD_bal_model_range)
plot(UD_list)

lambda = min_lambda_calc(UD_bal_model_range, 0.01)
UD_1percent = ppud(UD_bal_model_range, lambda)
UD_1percent$UD[[1]]

```

var0	<i>Variance(DEPRECATED)</i>
------	-----------------------------

Description

This function returns the population variance of a given vector.

Usage

```
var0(x)
```

Arguments

x a numeric vector.

Value

the population variance of a given vector

Examples

```

(x <- rnorm(5))
# var0() is deprecated. Use U_var()
# var0(x)

```

Index

- * **datasets**
 - unc_measures, [16](#)
- apply0 (apply_sweep), [2](#)
- apply_sweep, [2](#)
- cum_uncertainty, [3](#)
- drange, [4](#)
- flist_mad (unc_measures), [16](#)
- flist_range (unc_measures), [16](#)
- flist_var (unc_measures), [16](#)
- min_lambda_calc (unc_postprocess), [17](#)
- msweep (apply_sweep), [2](#)
- plotUDlist, [5](#)
- ppud (unc_postprocess), [17](#)
- print.UD_model (printUD), [6](#)
- print.UD_stage (printUD), [6](#)
- printUD, [6](#)
- scenario_uncertainty, [7](#)
- stage_uncertainty, [8](#)
- sweep0 (apply_sweep), [2](#)
- U_mad (unc_measures), [16](#)
- u_mad (unc_measures), [16](#)
- U_range (unc_measures), [16](#)
- u_range (unc_measures), [16](#)
- U_var (unc_measures), [16](#)
- u_var (unc_measures), [16](#)
- UD_ANOVA_model, [9](#)
- UD_ANOVA_stage, [10](#)
- UD_bal_model, [11](#)
- UD_bal_stage, [12](#)
- UD_cum_stage, [13](#)
- UD_model2stage, [14](#)
- UD_table (UDtable), [15](#)
- UDtable, [15](#)
- unc_measures, [16](#)
- unc_postprocess, [17](#)
- var0, [18](#)