

# Package ‘ZIprop’

May 7, 2026

**Type** Package

**Title** Permutations Tests and Performance Indicator for Zero-Inflated Proportions Response

**Version** 0.1.1

**Date** 2021-06-07

**Author** Melina Ribaud

**Maintainer** Melina Ribaud <melina.ribaud@gmail.com>

**Description** Permutations tests to identify factor correlated to zero-inflated proportions response. Provide a performance indicator based on Spearman correlation to quantify the part of correlation explained by the selected set of factors. See details for the method at the following preprint e.g.: <<https://hal.archives-ouvertes.fr/hal-02936779v3>>.

**URL** <https://gitlab.paca.inrae.fr/meribaud/ziprop>

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 3.5.0), rgenoud, purrr, data.table, parallel

**Suggests** markdown, knitr, ggplot2, ggrepel, ggthemes, kableExtra, stringr

**RoxygenNote** 7.1.1

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-06-09 13:20:02 UTC

## Contents

delta . . . . .	2
diffFactors . . . . .	3
equineDiffFactors . . . . .	3
example_data . . . . .	4

fact2mat . . . . .	5
indicator . . . . .	5
indicator_max . . . . .	6
model_matrix . . . . .	7
permDT . . . . .	8
scale_01 . . . . .	9
T_stat_discr . . . . .	10
T_stat_multi . . . . .	10
ZIprop . . . . .	11

<b>Index</b>	<b>12</b>
--------------	-----------

---

delta	<i>The scalar delta</i>
-------	-------------------------

---

### Description

Calculate the scalar delta. This parameter comes from the optimal Spearman's correlation when the rank of two vectors  $X$  and  $proba$  are equal except on a given set of indices. In our context, this set correspond to the zero-values of the vector  $proba$ .

### Usage

```
delta(X, proba)
```

### Arguments

$X$	a vector.
$proba$	a zero-inflated proportions response.

### Value

Delta the scalar Delta calculated for the vector  $x$  and the vector  $proba$ .

### Examples

```
X = rnorm(100)
proba = runif(100)
proba[sample(1:100,80)]=0
Delta = delta(X,proba)
print(Delta)
```

---

`diffFactors`*diffFactors*

---

**Description**

Data for the comparison of COVID-19 mortality in European and North American geographic entities

**Usage**

```
data(diffFactors)
```

**Format**

A data frame with 483 rows and 32 variables

**Details**

- `geographic_entity_receptor` are the entity receptor
- `geographic_entity_source` are the entity source
- `proba` is the probability that the receptor follows the mortality dynamics of the source
- other columns are the difference between factors

**Author(s)**

Melina Ribaud, Davide Martinetti and Samuel Soubeyrand

**References**

doi: [10.5281/zenodo.4769671](https://doi.org/10.5281/zenodo.4769671)

---

`equineDiffFactors`*equineDiffFactors*

---

**Description**

Equine Influenza dataset

**Usage**

```
data(equineDiffFactors)
```

**Format**

A data frame with 2256 rows and 8 variables

**Details**

- ID.source are the ID of source hosts
- ID.recep are the ID of receiver hosts
- y are the vector of transmission probabilities source -> receiver
- other columns are the factors

**Author(s)**

Melina Ribaud and Joseph Hughes

**References**

doi: [/10.5281/zenodo.4837560](https://doi.org/10.5281/zenodo.4837560)

---

example\_data

*Zero-inflated proportions dataset*

---

**Description**

A dataset example to test the package functions. The factor X1 to X5 and F1 to F5 are correlated to the responses y.

**Usage**

```
data(example_data)
```

**Format**

A data frame with 440 rows and 23 variables

**Details**

- ID.source are the ID of source hosts
- ID.recep are the ID of receiver hosts
- y are the vector of transmission probabilities source -> receiver
- X1 to X10 are continuous factor
- F1 to F10 are discrete factor

---

fact2mat	<i>Turn factor into multiple column</i>
----------	---

---

**Description**

Turns a factor with several levels into a matrix with several columns composed of zeros and ones.

**Usage**

```
fact2mat(x)
```

**Arguments**

x                    a vector.

**Value**

Columns with zeros and ones.

**Examples**

```
x = sample(1:3,100,replace = TRUE)
fact2mat(x)
```

---

indicator	<i>The performance indicator</i>
-----------	----------------------------------

---

**Description**

Calculate the indicator for a vector X and a zero-inflated proportions response proba.

**Usage**

```
indicator(X, proba)
```

**Arguments**

X                    a vector.  
proba                a zero-inflated proportions response.

**Value**

a scalar represents the performance indicator and the vector proba.

**Examples**

```
X = rnorm(100)
proba = runif(100)
proba[sample(1:100,80)]=0
print(indicator(X,proba))
```

---

indicator_max	<i>The max performance indicator</i>
---------------	--------------------------------------

---

**Description**

Search for the set of parameters that maximize the indicator (equivalent to Spearman correlation). For a given set of factors scaled between 0 and 1 and a zero-inflated proportions response.

**Usage**

```
indicator_max(  
  DT,  
  ColNameFactor,  
  ColNameWeight = "weight",  
  bounds = c(-10, 10),  
  max_generations = 200,  
  hard_limit = TRUE,  
  wait_generations = 50,  
  other_class = NULL  
)
```

**Arguments**

DT	a data table contains the factors and the response.
ColNameFactor	a char vector with the name of the selected factor.
ColNameWeight	a char with the name of the ZI response.
bounds	default is $[-10;10]$ . Upper and Lower bounds.
max_generations	default is 200 see <a href="#">genoud</a> for more information.
hard_limit	default is TRUE see <a href="#">genoud</a> for more information.
wait_generations	default is 50 see <a href="#">genoud</a> for more information.
other_class	a char vector with the name of other classes than numeric (factor or char).

**Value**

Return a list of two elements with the value of the indicator and the associate set of parameters (beta).

**Examples**

```
library(data.table)
data(example_data)
# For real cases increase max_generations and wait_generations
I_max = indicator_max(example_data,
names(example_data)[c(4:8, 14:18)],
ColNameWeight = "proba",
max_generations = 20,
wait_generations = 5)
print(I_max)
```

---

model\_matrix

*Construct Design Matrix*

---

**Description**

Creates a design matrix by expanding factors to a set of dummy variables.

**Usage**

```
model_matrix(DT, ColNameFactor, other_class)
```

**Arguments**

DT                    a data table contains the factors and the response.  
ColNameFactor        a char vector with the name of the selected factor.  
other\_class          a char vector with the name of other classes than numeric (factor or char).

**Value**

return the value.

**Examples**

```
library(data.table)
data(example_data)
m = model_matrix (example_data,
colnames(example_data)[-c(1:3)],
other_class = colnames(example_data)[14:23])
print(m)
```

permDT

*Permutations tests***Description**

Permutations tests to identify factor correlated to a zero-inflated proportions response. The statistic are the Spearman's correlation for numeric factor and mean by level for other factor.

**Usage**

```
permDT(
  DT,
  ColNameFactor,
  B = 1000,
  nclust = 1,
  ColNameWeight = "weight",
  ColNameRecep = "ID.recep",
  ColNameSource = "ID.source",
  seed = NULL,
  no_const = FALSE,
  num_class = ColNameFactor,
  other_class = NULL,
  multiple_test = FALSE,
  adjust_method = "none",
  alpha = 0.05
)
```

**Arguments**

DT	a data table contains the factors and the response.
ColNameFactor	a char vector with the name of the selected factor.
B	number of permutations (use at least B=1000 permutations to get a correct accuracy of the p-value.)
nclust	number of proc for parallel computation.
ColNameWeight	a char with the name of the ZI response.
ColNameRecep	colname of the column with the target names
ColNameSource	colname of the column with the contributor names
seed	vector with the seed for the permutations: size(seed)=B
no_const	FALSE for receiver block constraint for permutations: TRUE no constraint.
num_class	a char vector with the name of numeric factor.
other_class	a char vector with the name of other classes than numeric (factor or char).
multiple_test	useful option only for discrete factors: Set TRUE to calculate multiple tests.
adjust_method	p-values adjusted methods (default "none" ). c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none").
alpha	significant level (default 0.05).

**Value**

A data frame with two columns. One for the statistics and the other one for the p-value.

**Examples**

```
library(data.table)
data(example_data)
res = permDT (example_data,
  colnames(example_data)[c(4,10,14,20)],
  B = 10,
  nclust = 1,
  ColNameWeight = "y",
  ColNameRecep = "ID.recep",
  ColNameSource = "ID.source",
  seed = NULL,
  num_class = colnames(example_data)[c(4,10)],
  other_class = colnames(example_data)[c(14,20)])
print(res)
```

---

scale\_01

*Scale vector*

---

**Description**

Scale a vector between 0 and 1.

**Usage**

```
scale_01(x)
```

**Arguments**

x                    a vector.

**Value**

the scaled vector of x.

**Examples**

```
x = runif(100,-10,10)
x_scale = scale_01(x)
range(x_scale)
```

---

T_stat_discr	<i>Statistic for non-numeric factor tests</i>
--------------	---

---

**Description**

Statistic for non-numeric factor tests (same statistic as H-test).

**Usage**

```
T_stat_discr(permu, al)
```

**Arguments**

permu	the response vector.
al	the factor.

**Value**

the statistic.

**Examples**

```
permu = runif(100,-10,10)
al = as.factor(sample(1:3,100,replace=TRUE))
T_stat_discr(permu, al)
```

---

T_stat_multi	<i>Statistic for non-numeric factor multiple tests</i>
--------------	--

---

**Description**

Statistic for non-numeric factor multiple tests (difference in mean ranks).

**Usage**

```
T_stat_multi(permu, al)
```

**Arguments**

permu	the response vector.
al	the factor.

**Value**

the means difference of two levels for a discrete factor.

**Examples**

```
permu = runif(100,-10,10)
al = as.factor(sample(1:3,100,replace=TRUE))
T_stat_multi(permu, al)
```

---

Zlprop

*Zlprop: A package for Zero-Inflated Proportions data (Zlprop)*

---

**Description**

We propose a by block-permutation-based methodology (i) to identify factors (discrete or continuous) that are potentially significant, (ii) to define a performance indicator to quantify the percentage of correlation explained by the significant factors subset for Zero-Inflated Proportions data (Zlprop).

**References**

Melina Ribaud, Edith Gabriel, Joseph Hughes, Samuel Soubeyrand. Identifying potential significant factors impacting zero-inflated proportions data. 2020. hal-02936779

# Index

- \* **diffFactors**
  - diffFactors, 3
- \* **equineDiffFactors**
  - equineDiffFactors, 3
- \* **example\_data**
  - example\_data, 4
  
- delta, 2
- diffFactors, 3
  
- equineDiffFactors, 3
- example\_data, 4
  
- fact2mat, 5
  
- genoud, 6
  
- indicator, 5
- indicator\_max, 6
  
- model\_matrix, 7
  
- permDT, 8
  
- scale\_01, 9
  
- T\_stat\_discr, 10
- T\_stat\_multi, 10
  
- ZIprop, 11