

Package ‘resp’

October 16, 2017

Title Creates response models for whole datasets

Version 1.0.2

Description Given a data.frame and selecting fixed and random factors, returns a set of models that allow to select the significant effect of the fixed factors.

Depends R (>= 3.0.2)

Imports lme4, stats

Encoding UTF-8

Author Adrià Masip <adria@burgeon.cat>

Maintainer resp Team <webmaster@burgeon.cat>

License GPL (>= 2)

R topics documented:

resp-package	1
formula_from_vec	2
log.dataset	3
mod.check	3
mod.resp	4
name_range	5
print.mresp	6

resp-package	<i>Creates response models for whole datasets</i>
--------------	---

Description

Given a data.frame and selecting fixed and random factors, returns a set of models that allow to select the significant effect of the fixed factors.

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.

~~ An overview of how to use the package, including the most important functions ~~

Author(s)

Adrià Masip <adria@burgeon.cat>

Maintainer: resp Team <webmaster@burgeon.cat>

References

~~ Literature or other references for background information ~~

See Also

~~ Optional links to other man pages, e.g. ~~ <pkg> ~~

Examples

~~ simple examples of the most important functions ~~

formula_from_vec	<i>Formula string creation from vector elements</i>
------------------	---

Description

Given a vector, create a unique string from the elements.

Usage

```
formula_from_vec(x, start='~ ', mid=' + ', end='', as_formula=FALSE)
```

Arguments

x	vector to be transformed
start	initial string, DEFAULT: '~ '
mid	between-elements string, DEFAULT: ' + '
end	final string, DEFAULT: ''
as_formula	logical, return string as S formula, DEFAULT: FALSE

Value

Returns single string of concatenated elements

log.dataset	<i>log-transform data.frame</i>
-------------	---------------------------------

Description

given a whole dataset in data.frame format, log-transform the values avoiding NAs and zeros, so no -Inf or NaN is created

Usage

```
log.dataset(data, columns = 1:dim(data)[2])
```

Arguments

data	data.frame to be transformed
columns	columns to be transformed, DEFAULT: whole data.frame

Value

Returns the same dataframe with log-transformed values

mod.check	<i>check models to recover performance information</i>
-----------	--

Description

After a mresp object is created, check for fixed factors significance and unvalidated results, mod.check can also be used to compare the performance of the created mixed models

Usage

```
mod.check(models, omit_NA=TRUE)
```

Arguments

models	mresp object to be analyzed
omit_NA	logical, compare data without NA values, DEFAULT: TRUE

Value

Returns the given mresp object with new check_out element inside each response variable

mod.resp

Create response lmer models for a whole data.frame

Description

Given the fixed and random factors of a data.frame, creates a list of 5 mixed models for each response variable, so the best fit can be selected

Usage

```
mod.resp(data, fixed, random, r_group, exclude,
omit_NA = TRUE, fixed_interaction = TRUE, check_models = TRUE)
```

Arguments

data	data.frame to be analyzed
fixed	vector of column names to be used as fixed factors
random	vector of column names to be used as random factors
r_group	vector of column names to be used as random grouping factors
exclude	vector of column names to be excluded from the response analysis
omit_NA	logical, DEFAULT TRUE, avoid using NAs from data
fixed_interaction	logical, DEFAULT TRUE, check interaction from fixed factors
check_models	logical, DEFAULT TRUE, create check_out table for each response variable
lmer_warnings	logical, DEFAULT FALSE, display lmer() construction warnings

Value

Returns mresp object with a list of response variables sorted alphabetically, with 5 models each and a comparison between them (checkout)

resp_1	First response variable
resp_...	Other response variables
resp_n	Last response variable

References

lmer, stats

Examples

```

n <- 500L
tr <- c('T1', 'T2', 'T3')
sp <- c('S1', 'S2', 'S3')
gr <- c('A', 'B')
F1 <- c(); F2 <- c(); F3 <- c()
R1 <- c(); R2 <- c(); R3 <- c()
for (i in 1:n) {
  F1[i] <- tr[round(i/n*3,0)]
  F2[i] <- sp[round(runif(1L,1L,3L),0)]
  F3[i] <- gr[round(runif(1L,1L,2L),0)]
  R1[i] <- rnorm(1,10,2)+runif(1L)+(which(tr==F1[i])*3)
  R2[i] <- rnorm(1,10,2)+runif(1L)+(which(sp==F2[i])*10)
  R3[i] <- rnorm(1,600,20)+runif(1L)+(which(tr==F1[i])*20)
}
table(F1,F2)

x <- data.frame(
  Treatment=F1,
  Specie=F2,
  Group=F3,
  Rand=runif(n),
  Height=R1,
  Diameter=R2,
  Number_leaves=R3,
  other=runif(n)
)
# rm(n,tr,sp,gr,F1,F2,F3)
a <- mod.resp(data = x, fixed = c('Treatment','Specie'), random='Rand',
              r_group = c('Group'), exclude='other', lmer_warnings=TRUE,
              choose_models=FALSE, check_models=FALSE)

print(a)

```

name_range

*Numeric data categorization between user-given ranges***Description**

Returns 'High', ['Mid[_n]'] or 'Low' based on range values for group limits

Usage

```
name_range(x, range)
```

Arguments

x	numeric set to be transformed, must be vector or data.frame
range	limits of the groups to be created

Value

Returns `vector` or `data.frame` of transformed elements.

<code>print.mresp</code>	<i>Print method for <code>mresp</code> objects</i>
--------------------------	--

Description

Prints a resume table of the relations between models based on logLikelihood ratio