Package 'SensMixed'

March 14, 2018

Type Package

Title Analysis of Sensory and Consumer Data in a Mixed Model Framework

Version 2.1-0

Depends R (>= 3.0.0), stats, methods

Imports Hmisc, plyr, doBy, xtable, reshape2, ggplot2, shiny, shinyBS, lme4 (>= 1.1), MASS

Suggests pbkrtest, parallel, estimability

Description Functions that facilitate analysis of Sensory as well as Consumer data within a mixed effects model framework. The so-called mixed assessor models, that correct for the scaling effect are implemented. The generation of the d-tilde plots forms part of the package. The shiny application provides a GUI for the functionalities.

LazyData TRUE

License GPL (>= 2)

NeedsCompilation no

Author Alexandra Kuznetsova [aut], Per Bruun Brockhoff [aut, ths, cre], Rune Haubo Bojesen Christensen [aut]

Maintainer Per Bruun Brockhoff <perbb@dtu.dk>

Repository CRAN

Date/Publication 2018-03-14 10:35:48 UTC

R topics documented:

`conjoint` *Conjoint analysis within a mixed effects model framework.*

Description

Performs conjoint analysis within a mixed effects model framework. This function is then used in the conjoint analysis of the ConsumerCheck software tool.

Usage

```
conjoint(structure = 1, data, response, fixed, random, facs,
         corr = FALSE, alpha.random = 0.1, alpha.fixed = 0.05)
```

Arguments

`structure`	numerical, takes values in c(1,2,3). Specifies the complexity of the conjoint model. **structure = 1** mixed effects model includes fixed main effects. Random effects consist of random consumer effect and interaction between consumer and the main effects. **structure = 2** mixed effects model includes main effects and all 2-factor interactions. Random effects consist of consumer effect and interaction between consumer and all fixed effects (both main and interaction ones). **structure = 3** This is a full factorial model with all possible fixed and random effects (i.e. including all main effects and all higher-way interactions). The automated reduction in random part is followed by an automated reduction in fixed part.
`data`	a data frame.
`response`	a vector. Names of the variables that correspond to the liking scores.
`fixed`	a list. first element is a vector with a name Product includes names of the design variables. second element with a name Consumer includes names of the consumer characteristics variables
`random`	a string with the name for a consumer variable
`facs`	a vector with the names of the variables that need to be considered as factors
`corr`	a logical vaue. TRUE if the correlations between random effects are included in the model. FALSE if correlations between random effects are set to 0.
`alpha.random`	significance level for elimination of the random part (for LRT test)
`alpha.fixed`	significance level for elimination of the fixed part (for F test)

Details

Conjoint analysis (Green and Rao 1971; Green and Srinivasan 1978) is a method for analysing the effects of design factors and consumer characteristics on consumer likings. A common approach is to analyse it in a mixed effects model framework, where random effects consist of consumer effect and interactions between consumer effects and design factors, and fixed effects consist of design factors and consumer characteristics and possibly interactions between them.

Value

`rand.table` data frame with value of Chi square statistics, p-values for the likelihood ratio test for random effects. The order of elimination of non-significant random effects for strucutre = 3

`anova.table` data frame with tests for whether the model fixed terms are significant (Analysis of Variance)

`lsmeans.table` Least Squares Means data frame with p-values and Confidence intervals

`diffs.lsmeans.table` Differences of Least Squares Means data frame with p-values and Confidence intervals. The multiplicity correction for multiple comparisons tests uses Bonferroni method

`residuals` These are the residuals of the final model (for structure = 3, the reduced one)

`residualsFixed` These are the residuals that are extracted from a fixed effects model with a saturated fixed structure (main effects and all higher-way interactions form fixed part) and one fixed Consumer effect.

Note

The tests for the random effects use likelihood ratio tests while the tests for the fixed effects use the F-test with Satterthwaites approximation to degrees of freedom. The automated reduction in the fixed part uses the principle of marginality, i.e. the highest order interactions are tested first: if they are significant, the lower order effects are not eliminated even if being non-significant.

Author(s)

Alexandra Kuznetsova, Per Bruun Brockhoff

References

Green P, Rao V (1971). Conjoint measurement for quantifying judgemental data. Journal of Marketing Research, 8, 355 - 363

Green P, Srinivasan V (1978). Conjoint analysis in consumer research: Issues and outlook. Journal of Consumer Research, 5, 103 - 123

See Also

sensmixed, SensMixedUI

Examples

```
## check with the  ham
response <- c("Liking")
fixed <- list(Product=c("Product", "Information"), Consumer="Gender")
random <- c("Consumer")
facs <- c("Consumer", "Product", "Information", "Gender")

res.ham <- conjoint(structure=3, ham, response, fixed, random, facs)
res.ham
res.ham$Liking
```

`convertToFactors` *converts variables of the data frame to factors*

Description

the user specifies which variables he/she would like to consider as factors, the functions converts them to factors

Usage

```
convertToFactors(data, facs)
```

Arguments

`data`	data frame
`facs`	vector with names of variables that the user would like to convert to factors

Value

returns the same data frame as in the input but with the specified variables converted to factors

Author(s)

Alexandra Kuznetsova

Examples

```
library(SensMixed)

str(ham)

ham <- convertToFactors(ham, c("Gender"))

str(ham)
```

ham *Conjoint study of dry cured ham*

Description

One of the purposes of the study was to investigate the effect of information given to the consumers measured in hedonic liking for the hams. Two of the hams were Spanish and two were Norwegian, each origin representing different salt levels and different aging time. The information about origin was given in such way that both true and false information was given. essentially a 4*2 design with 4 samples and 2 information levels. A total of 81 Consumers participated in the study.

Usage

```
ham
```

Format

`Consumer` factor with 81 levels: numbering identifying consumers

`Product` factor with four levels

`Informed.liking` numeric: hedonic liking for the products

`Information` factor with two levels

`Gender` factor with two levels (gender)

`Age` numeric: age of Consumer

References

"Alternative methods for combining design variables and consumer preference with information about attitudes and demographics in conjoint analysis" . T. Naes, V.Lengard, S. Bolling Johansen, M. Hersleth

Examples

```
## check with the  ham
response <- c("Liking")
fixed <- list(Product=c("Product", "Information"), Consumer="Gender")
random <- c("Consumer")
facs <- c("Consumer", "Product", "Information", "Gender")

res.ham <- conjoint(structure=1, ham, response, fixed, random, facs)
```

plot *function creates plots for the sensmixed object*

Description

function creates barplots for the square roots of F statistics and square roots of chi square values for all attributes

Usage

```
## S3 method for class 'sensmixed'
 plot(x, mult = FALSE, dprime = FALSE, sep = FALSE,
                              cex = 2,  interact.symbol = ":",
                            isRand = TRUE, isScaling = TRUE, stacked = TRUE, ...)
```

Arguments

x	object of class sensmixed
mult	logical. Should multiple plots be plotted, that is barplots for each effect in a separate plot
dprime	logical. Should multiattribute plot for product effects use average squared dprimes instead of square root of F statistics
sep	logical. If TRUE then separate plot is plotted for each effect (mult argument should be then also TRUE)
cex	The magnification to be used
interact.symbol	The symbol to be used for the interaction effects
isRand	logical. Whether to plot tests of the random effects
isScaling	logical. Whether to plot the scaling factor if present
stacked	logical. Whether bars should be stacked
...	other potential arguments.

Value

NULL is returned

Author(s)

Alexandra Kuznetsova

Examples

```
res <- sensmixed(c("Coloursaturation", "Colourbalance", "Noise"),
                 prod_effects=c("TVset", "Picture"),
                 assessor="Assessor", data=TVbo, MAM=TRUE,
                 control=list(reduce.random=FALSE))
plot(res, isRand = TRUE)
plot(res, isRand = FALSE)
plot(res, stacked = FALSE, mult = TRUE)
plot(res, isRand = FALSE, stacked = FALSE, interact.symbol = " x ")
```

`plot.conjoint` *plots the post-hoc for the conjoint object*

Description

plots the least squares means and differences of least squares means together with the confidence intervals for the fixed effects

Usage

```
## S3 method for class 'conjoint'
 plot(x, main = NULL, cex = 1.4,
                            which.plot = c("LSMEANS", "DIFF of LSMEANS"),
                            test.effs = NULL, ...)
```

Arguments

`x`	object of class conjoint
`main`	string. Title for the plots
`cex`	A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default
`which.plot`	type of plot to be drawn
`test.effs`	name of the effext for which to draw the plots
`...`	other potential arguments.

Value

returns NULL

Author(s)

Alexandra Kuznetsova

Examples

```
#convert some variables to factors in Tham
response <- c("Liking")
fixed <- list(Product=c("Product", "Information"), Consumer="Gender")
random <- c("Consumer")
facs <- c("Consumer", "Product", "Information", "Gender")

res.ham <- conjoint(structure=3, ham, response, fixed, random, facs)
```

`plotLSMEANS` *plots bars for the LSMEANS or differences of LSMEANS*

Description

plots bars for the LSMEANS or differences of LSMEANS for product factors and confidence intervals

Usage

```
plotLSMEANS(table, response,
                        which.plot=c("LSMEANS", "DIFF of LSMEANS"),
                        main = NULL, cex = 1.4, effs = NULL, mult = TRUE)
```

Arguments

`table`	data table containing LSMEANS/ DIFFLSMEANS table from the step function of the lmerTest package
`response`	vector with the name of the attribute, for which the LSMEANS / DIFFLSMEANS are calculated
`which.plot`	name, indicating the type of plot to generate.
`main`	name of the title for the plot
`cex`	cex for representing the plot (UNUSED?..)
`effs`	vector with the names for the effects, for which to plot the LSMEANS / DIFFLSMEANS
`mult`	logical. TRUE means plot LSMEANS for all effects in one layout
`...`	other potential arguments.

Value

barplots created via ggplot2 package

Author(s)

Alexandra Kuznetsova, Per Bruun Brockhoff, Rune Haubo Bojesen Christensen

saveToDoc *save the result in tables into a doc file for sensmixed or conjoint objects*

Description

save the tests for the random and fixed effects into a doc file for sensmixed or conjoint objects

Usage

```
saveToDoc(x, file = NA, bold = FALSE, append = TRUE, type = "html", typeEffs = 1)
```

Arguments

x	object of class sensmixed or conjoint.
file	a character string naming the file to write to, or NULL to stop sink-ing.
bold	logical. Should the significance be in bold text instead of the stars. The default is FALSE
append	logical. If TRUE, output will be appended to file; otherwise, it will overwrite the contents of file.
type	type of output as in xtable. Either "html" or "latex"
typeEffs	one of the numbers in c(1,2,3,4). 1: save results for the random effects in a table 2: save results for the fixed effects in a table 3: save results for the scaling effects in a table 4: save results for all effects in a table

Author(s)

Alexandra Kuznetsova

Examples

```
## Not run:
res <- sensmixed(c("Coloursaturation", "Colourbalance"),
                 prod_effects=c("TVset"),
                 assessor="Assessor", data=TVbo)

saveToDoc(res, file = "C:/Desktop/output.doc")

## End(Not run)
```

sensmixed *Analysis of sensory data within a mixed effects model framework*

Description

Constructs mixed effects models for each of the selected by a user attributes. By default the largest possible models (that contain all possible interactions in fixed and random parts) are fitted. The complexity of the fitted models can be changed. Non-significant random effects are eliminated (by default). The likelihood ratio test (LRT) is used for testing the random terms, F-type hypothesis test is used for testing the terms. The type of the model and the type of the analysis can be changed with the `control` argument (see `sensmixedControl()`)

Usage

```
sensmixed(attributes, prod_effects, assessor,
          replication = NULL, data, product_structure = 3,
          error_structure ="ASS-REP", MAM = TRUE,
          control = sensmixedControl())
```

Arguments

`attributes`	a vector with names of sensory attributes
`prod_effects`	a vector with the names of the variables related to the product
`replication`	a character with the name of the replication column in the data, if present
`assessor`	a characthe with the name of the column in the data that represents assessors
`data`	a data frame (data from sensory studies)
`product_structure`	numeric, takes values in in c(1, 2, 3). Specifies the complexity of the fixed part (product effects) of the mixed effects models for all attributes. **product_structure = 1** only main effects **product_structure = 2** main effects and 2-way interactions **product_structure = 3** all main effects and all possible interaction
`error_structure`	character, takes values in c("ONLY-ASS", "ASS-REP"). Specifies the complexity of the random part of the mixed effects models for all attributes. **error_structure = "ONLY-ASS"** assessor effect and all possible interactions between assessor and product effects **error_structure = "ASS-REP"** assessor and replicate effect and interaction between them and interaction between them and product effects
`MAM`	logical. if TRUE then mixed assessor models (MAM) are fitted for the selected attributes (see Brockhoff, P. B., Schlich, P., & Skovgaard, I. (2015))
`control`	a list (of class `sensmixedControl()`) containing control parameters. See the `sensmixedControl()` documentation for details.

Value

FCHi	matrix with Chi square values from LRT test and F values form F-type test for the selected attributes
pvalue	matrix withp-values for all effects for the selected attributes

Author(s)

Alexandra Kuznetsova, Per Bruun Brockhoff, Rune Haubo Bojesen Christensen

References

Brockhoff, P. B., Schlich, P., & Skovgaard, I. (2015). Taking individual scaling differences into account by analyzing profile data with the mixed assessor model. Food Quality and Preference, 39 , 156-166.

Kuznetsova, A., Christensen, R. H., Bavay, C., & Brockhoff, P. B. (2015). Automated mixed ANOVA modeling of sensory and consumer data. Food Quality and Preference, 40, Part A, 31 38. URL: http://www.sciencedirect.com/science/ article/pii/S0950329314001724. doi:http://dx.doi.org/10.1016/j.foodqual.2014.08.004.

See Also

sensmixedControl, conjoint, SensMixedUI

Examples

```
## import SensMixed package
library(SensMixed)

## convert some variables to factors in TVbo
TVbo <- convertToFactors(TVbo, c("Assessor", "Repeat", "Picture"))

## run automated selection process
res <- sensmixed(c("Coloursaturation", "Colourbalance"),
prod_effects = c("TVset", "Picture"),
assessor="Assessor", data=TVbo, MAM=TRUE)

res


## run MAManalysis function
res_MAM <- sensmixed(c("Coloursaturation", "Colourbalance"),
                prod_effects=c("TVset"), replication="Repeat",
                assessor="Assessor", data=TVbo, control = list(MAM_balanced=TRUE))
## print is not yet implemented
## get anova part
res_MAM[[3]][,,1]

## compare with the general implementation
```

```
res <- sensmixed(c("Coloursaturation", "Colourbalance"),
                 prod_effects=c("TVset"),
                 assessor="Assessor", data=TVbo, MAM=TRUE,
                 control = list(reduce.random=FALSE))
res$fixed

## Not run:
res <- sensmixed(names(TVbo)[5:(ncol(TVbo) - 1)],
                 prod_effects=c("TVset", "Picture"),
                 assessor="Assessor",
                 data=TVbo)


plot F and Chi square values
plot(res)

## End(Not run)
```

`sensmixedControl` *Control of sensmixed function*

Description

Construct control structures for `sensmixed` function.. All arguments have defaults, and can be grouped into

- mixed assessor model (MAM) parameters (are only used if MAM = TRUE in the `sensmixed` function)
- model analysis specifications

Usage

```
sensmixedControl(MAM_mult_scaling = FALSE, MAM_oneway_rand = FALSE,
                                MAM_balanced = FALSE, MAM_adjusted = FALSE,
                                MAM_alpha_conditional = 1,
                                calc_post_hoc = FALSE, parallel = FALSE,
                                reduce.random=TRUE, alpha.random = 0.1,
                                alpha.fixed = 0.05, interact.symbol = ":",
                                keep.effs = NULL)
```

Arguments

`MAM_mult_scaling`
: logical. Whether multiple scaling should be used. This option is not fully investigated (recommened is FALSE)

`MAM_oneway_rand`
: logical. Whether there should be just one `prod_effect` (in a multi-factorial product structure setting this means the highest order interaction) as part of the random part in MAM

MAM_balanced	logical. if MAManalysis function should be called (scaling correction). Can only be used for one product effect, no replications and balanced data.
MAM_adjusted	logical. should MAM be adjusted for the scaling (is only used if MAM_balanced = TRUE)
MAM_alpha_conditional	logical. scaling should be part of the model in case its p-value is less than MAM_alpha_conditional (is only used if MAM_balanced = TRUE)
calc_post_hoc	logical. Should the post hoc analysis be performed for the attributes
reduce.random	logical. Eliminate non-significant random effects according to alpha.random or not. The default is TRUE
keep.effs	a vector with the names which effects should be kept in a model (even if non-significant). By default the assessor effect and interaction between assessor and highest order product effects are always kept.
alpha.random	significance level for elimination of the random part (for LRT test)
alpha.fixed	significance level for elimination of the fixed part (for F test)
interact.symbol	symbol for the indication of the interaction between effects (for presenting in tables and plots). the default is ":".
parallel	logical. Should the computation be done in parallel for the attributes (saves computational time). the default is FALSE

Value

The sensmixedControl function returns a list containing

1. mixed assessor model fitting related control parameters (are used only if MAM=TRUE in the sensmixed function).
2. control parameters for the analysis.

See Also

sensmixed

Examples

```
str(sensmixedControl())
```

SensMixedUI *Graphical User interface for the SensMixed package*

Description

launches a shiny application that provides the graphical user interface (GUI) for the functions contained in the SensMixed package. Application also includes such crucial functionalities as importing the data in different formats, presenting results in tables and plots as well as saving them.

Usage

```
SensMixedUI()
```

Author(s)

Alexandra Kuznetsova, Per Bruun Brockhoff, Rune Haubo Bojesen Christensen

See Also

sensmixed, conjoint

Examples

```
## Not run:
library(SensMixed)
SensMixedUI()

## End(Not run)
```

TVbo *TV dataset*

Description

The TVbo dataset comes from Bang and Olufsen company. The main purpose was to test products, specified by two attributes Picture and TVset. 15 different response variables (characteristics of the product) were assessed by trained panel list.

Usage

```
TVbo
```

Format

`Assessor` factor: numbering identifying assessors

`TVset` factor: attribute of the product

`Picture` factor: attribute of the product

`15 Characteristics of the product` numeric variables: Coloursaturation, Colourbalance, Noise, Depth, Sharpness, Lightlevel, Contrast, Sharpnessofmovement, Flickeringstationary, Flickeringmovement, Distortion, Dimglasseffect, Cutting, Flossyedges, Elasticeffect

Source

Bang and Olufsen company

Examples

```
## import SensMixed package
library(SensMixed)

## convert some variables to factors in TVbo
TVbo <- convertToFactors(TVbo, c("Assessor", "Repeat", "Picture"))

## run automated selection process
res <- sensmixed(c("Coloursaturation", "Colourbalance"),
prod_effects = c("TVset", "Picture"),
assessor="Assessor", data=TVbo, MAM=FALSE)
res
```

Index