

Package ‘vcdExtra’

May 8, 2012

Type Package

Title vcd additions

Version 0.5-3

Date 2012-03-07

Author@R c(person(given = “Michael”, family = “Friendly”, role=c(“aut”, “cre”), email=“friendly@yorku.ca”), person(given = “Heather”, family = “Turner”, role=“ctb”), person(given = “Achim”, family = “Zeileis”, role=“ctb”), person(given = “Duncan”, family = “Murdoch”, role=“ctb”), person(given = “David”, family = “Firth”, role=“ctb”))

Author Michael Friendly with Heather Turner, Achim Zeileis, Duncan Murdoch and David Firth

Maintainer Michael Friendly <friendly@yorku.ca>

Depends R (>= 2.10), vcd, gnm

Suggests ca, gmodels, Fahrmeir, effects, VGAM, plyr, rgl, lmtree, MASS, nnet

Description Provides additional data sets, methods and documentation to complement the vcd package for Visualizing Categorical Data and the gnm package for Generalized Nonlinear Models. In particular, vcdExtra extends mosaic, assoc and sieve plots from vcd to handle glm() and gnm() models and adds a 3D version in mosaic3d.

License GPL (>= 2)

URL <http://CRAN.R-project.org/package=vcdExtra>

LazyLoad yes

LazyData yes

Repository CRAN

Date/Publication 2012-05-08 04:07:15

R topics documented:

vcdExtra-package	3
Abortion	4
Alligator	5
Bartlett	7
Caesar	8
Cancer	9
collapse.table	10
Crossings	11
datasets	12
DaytonSurvey	13
Detergent	14
Dyke	15
expand.dft	16
Fungicide	18
Gilby	19
GKgamma	20
glm1ist	21
GSS	22
Hauser79	23
Heart	25
Heckman	26
Hoyt	27
ICU	29
JobSat	31
Kway	31
loddsratio	33
Mammograms	38
meanResiduals	39
Mental	40
Mobility	41
modFit	42
mosaic.glm	43
mosaic3d	46
print.Kappa	49
split3d	49
summarise	51
TV	52
Vietnam	54
Yamaguchi87	55

Description

This package provides additional data sets, documentation, and a few functions designed to extend the vcd package for Visualizing Categorical Data and the gnm package for Generalized Nonlinear Models. In particular, vcdExtra extends mosaic, assoc and sieve plots from vcd to handle glm() and gnm() models and adds a 3D version in [mosaic3d](#).

Details

Package:	vcdExtra
Type:	Package
Version:	0.5-3
Date:	2012-03-07
License:	GPL version 2 or newer
LazyLoad:	yes

The main purpose of this package is to serve as a sandbox for introducing extensions of mosaic plots and related graphical methods that apply to loglinear models fitted using glm() and related, generalized nonlinear models fitted with gnm() in the [gnm-package](#) package. The method [mosaic.glm](#) extends the [mosaic.loglm](#) method in the vcd package to this wider class of models. [mosaic3d](#) introduces a 3D generalization of mosaic displays using the rgl package.

In addition, there are several new data sets, a new vignette,

vcd-tutorial Working with categorical data with R and the vcd package, `vignette("vcd-tutorial", package = "vcdExtra")`

and a few functions for manipulating categorical data sets and working with models for categorical data.

A new class, [glmList](#), is introduced for working with collections of glm objects, e.g., [Kway](#) for fitting all K-way models from a basic marginal model, and [summarise](#) for brief statistical summaries of goodness-of-fit for a collection of models.

For square tables with ordered factors, [Crossings](#) supplements the specification of terms in model formulas using [Symm](#), [Diag](#), [Topo](#), etc. in the [gnm-package](#).

Some of these extensions may be migrated into vcd or gnm.

A collection of demos is included to illustrate fitting and visualizing a wide variety of models:

mental-glm Mental health data: mosaics for glm() and gnm() models

occStatus Occupational status data: Compare mosaic using `expected=` to `mosaic.glm`

ucb-glm UCBA admissions data: Conditional independence via `loglm()` and `glm()`

vision-quasi VisualAcuity data: Quasi- and Symmetry models

yaish-unidiff Yaish data: Unidiff model for 3-way table

Wong2-3 Political views and support for women to work (U, R, C, R+C and RC(1) models)

Wong3-1 Political views, support for women to work and national welfare spending (3-way, marginal, and conditional independence models)

housing Visualize glm(), multinom() and polr() models from example(housing, package="MASS")

Use `demo(package="vcdExtra")` for a complete current list.

The `vcdExtra` now contains a large number of data sets illustrating various forms of categorical data analysis and related visualizations, from simple to advanced. Use `data(package="vcdExtra")` for a complete list, or `datasets(package="vcdExtra")` for an annotated one showing the `class` and `dim` for each data set.

Author(s)

Michael Friendly

Maintainer: Michael Friendly <friendly AT yorku.ca>

References

Friendly, M. *Visualizing Categorical Data*, Cary NC: SAS Institute, 2000. Web materials: <http://www.datavis.ca/books/vcd/>.

Meyer, D.; Zeileis, A. & Hornik, K. The Strucplot Framework: Visualizing Multi-way Contingency Tables with `vcd` *Journal of Statistical Software*, 2006, **17**, 1-48. Available in R via `vignette("strucplot", package = "vcd")`.

Turner, H. and Firth, D. *Generalized nonlinear models in R: An overview of the gnm package*, 2007, <http://eprints.ncrm.ac.uk/472/>. Available in R via `vignette("gnmOverview", package = "gnm")`.

See Also

[gnm-package](#), for an extended range of models for contingency tables

[mosaic](#) for details on mosaic displays within the strucplot framework.

Examples

```
example(mosaic.glm)
```

```
demo("mental-glm")
```

Abortion

Abortion Opinion Data

Description

Opinions about abortion classified by gender and SES

Usage

```
data(Abortion)
```

Format

A 3-dimensional array resulting from cross-tabulating 3 variables for 1100 observations. The variable names and their levels are:

No	Name	Levels
1	Sex	"Female", "Male"
2	Status	"Lo", "Hi"
3	Support_Abortion	"Yes", "No"

Details

The combinations of Sex and Status represent four independent samples, having fixed Sex-Status marginal totals. Thus the Sex:Status association must be included in any loglinear model. Support_Abortion is a natural response variable.

Source

Christensen, R. (1990). *Log-Linear Models*, New York, NY: Springer-Verlag, p. 92, Example 3.5.2.

Christensen, R. (1997). *Log-Linear Models and Logistic Regression*, New York, NY: Springer, p. 100, Example 3.5.2.

Examples

```
data(Abortion)

# example goes here
ftable(Abortion)
mosaic(Abortion, shade=TRUE)

# stratified by Sex
fourfold(aperm(Abortion, 3:1))
# stratified by Status
fourfold(aperm(Abortion, c(3,1,2)))
```

Alligator

Alligator Food Choice

Description

The Alligator data, from Agresti (2002), comes from a study of the primary food choices of alligators in four Florida lakes. Researchers classified the stomach contents of 219 captured alligators into five categories: Fish (the most common primary food choice), Invertebrate (snails, insects, crayfish, etc.), Reptile (turtles, alligators), Bird, and Other (amphibians, plants, household pets, stones, and other debris).

Usage

```
data(Alligator)
```

Format

A frequency data frame with 80 observations on the following 5 variables.

lake a factor with levels George Hancock Oklawaha Trafford

sex a factor with levels female male

size alligator size, a factor with levels large (>2.3m) small (<=2.3m)

food primary food choice, a factor with levels bird fish invert other reptile

count cell frequency, a numeric vector

Details

The table contains a fair number of 0 counts.

food is the response variable. fish is the most frequent choice, and often taken as a baseline category in multinomial response models.

Source

Agresti, A. (2002). *Categorical Data Analysis*, New York: Wiley, 2nd Ed., Table 7.1

Examples

```
data(Alligator)

# change from frequency data.frame to table
allitable <- xtabs(count~lake+sex+size+food, data=Alligator)
# Agresti's Table 7.1
structable(food~lake+sex+size, allitable)

plot(allitable, shade=TRUE)
# mutual independence model
mosaic(~food+lake+size, allitable, shade=TRUE)
# food jointly independent of lake and size
mosaic(~food+lake+size, allitable, shade=TRUE, expected=~lake:size+food)

if (require(nnet)) {
# multinomial logit model
mod1 <- multinom(food ~ lake+size+sex, data=Alligator, weights=count)
}
```

Bartlett

Bartlett data on plum root cuttings

Description

In an experiment to investigate the effect of cutting length (two levels) and planting time (two levels) on the survival of plum root cuttings, 240 cuttings were planted for each of the 2 x 2 combinations of these factors, and their survival was later recorded.

Bartlett (1935) used these data to illustrate a method for testing for no three-way interaction in a contingency table.

Usage

```
data(Bartlett)
```

Format

A 3-dimensional array resulting from cross-tabulating 3 variables for 960 observations. The variable names and their levels are:

No	Name	Levels
1	Alive	"Alive", "Dead"
2	Time	"Now", "Spring"
3	Length	"Long", "Short"

Source

Hand, D. and Daly, F. and Lunn, A. D. and McConway, K. J. and Ostrowski, E. (1994). *A Handbook of Small Data Sets*. London: Chapman & Hall, p. 15, # 19.

References

Bartlett, M. S. (1935). Contingency Table Interactions *Journal of the Royal Statistical Society*, Supplement, 1935, 2, 248-252.

Examples

```
data(Bartlett)

# example goes here
fourfold(Bartlett, mfrow=c(1,2))

mosaic(Bartlett, shade=TRUE)
```

 Caesar

Risk Factors for Infection in Caesarian Births

Description

Data from infection from birth by Caesarian section, classified by Risk (two levels), whether Antibiotics were used (two levels) and whether the Caesarian section was Planned or not. The outcome is Infection (three levels).

Usage

```
data(Caesar)
```

Format

A 4-dimensional array resulting from cross-tabulating 4 variables for 251 observations. The variable names and their levels are:

No	Name	Levels
1	Infection	"Type 1", "Type 2", "None"
2	Risk	"Yes", "No"
3	Antibiotics	"Yes", "No"
4	Planned	"Yes", "No"

Details

Infection is regarded as the response variable here. There are quite a few 0 cells here, particularly when Risk is absent and the Caesarian section was unplanned. Should these be treated as structural or sampling zeros?

Source

Fahrmeir, L. & Tutz, G. (1994). *Multivariate Statistical Modelling Based on Generalized Linear Models* New York: Springer Verlag.

See Also

[caesar](#) for the same data recorded as a frequency data frame with other variables.

Examples

```
data(Caesar)
#display table; note that there are quite a few 0 cells
structable(Caesar)

# baseline model, Infection as response
Caesar.mod0 <- loglm(~Infection + (Risk*Antibiotics*Planned), data=Caesar)
```

```

# NB: Pearson chisq cannot be computed due to the 0 cells
Caesar.mod0

mosaic(Caesar.mod0, main="Baseline model")

# Illustrate handling structural zeros
zeros <- 0+ (Caesar >0)
zeros[1,,1,1] <- 1
structable(zeros)

# fit model excluding possible structural zeros
Caesar.mod0s <- loglm(~Infection + (Risk*Antibiotics*Planned), data=Caesar,
start=zeros)
Caesar.mod0s

anova(Caesar.mod0, Caesar.mod0s, test="Chisq")

mosaic (Caesar.mod0s)

# what terms to add?
add1(Caesar.mod0, ~.^2, test="Chisq")

# add Association of Infection:Antibiotics
Caesar.mod1 <- update(Caesar.mod0, ~.+Infection:Antibiotics)
anova(Caesar.mod0, Caesar.mod1, test="Chisq")

mosaic(Caesar.mod1, gp=shading_Friendly, main="Adding Infection:Antibiotics")

```

Cancer

Survival of Breast Cancer Patients

Description

Three year survival of 474 breast cancer patients according to nuclear grade and diagnostic center.

Usage

```
data(Cancer)
```

Format

A 3-dimensional array resulting from cross-tabulating 3 variables for 474 observations. The variable names and their levels are:

No	Name	Levels
1	Survival	"Died", "Surv"
2	Grade	"Malignant", "Benign"
3	Center	"Boston", "Glamorgan"

Source

Lindsey, J. K. (1995). Analysis of Frequency and Count Data Oxford, UK: Oxford University Press. p. 38, Table 2.5.

Whittaker, J. (1990) Graphical Models in Applied Multivariate Statistics New York: John Wiley and Sons, p. 220.

Examples

```
data(Cancer)

# example goes here
```

collapse.table	<i>Collapse levels of a table</i>
----------------	-----------------------------------

Description

Collapse (or re-label) variables in a a contingency table or ftable by re-assigning levels of the table variables

Usage

```
collapse.table(table, ...)
```

Arguments

table	A table or ftable object
...	A collection of one or more assignments of factors of the table to a list of levels

Details

Each of the ... arguments must be of the form `variable = levels`, where `variable` is the name of one of the table dimensions, and `levels` is a character or numeric vector of length equal to the corresponding dimension of the table.

Value

A `xtabs` and `table` objects, representing the original table with one or more of its factors collapsed or rearranged into other levels.

Author(s)

Michael Friendly

See Also

[expand.dft](#)

[margin.table](#) "collapses" a table in a different way, by summing over table dimensions.

Examples

```
# create some sample data in table form
sex <- c("Male", "Female")
age <- letters[1:6]
education <- c("low", 'med', 'high')
data <- expand.grid(sex=sex, age=age, education=education)
counts <- rpois(36, 100)
data <- cbind(data, counts)
t1 <- xtabs(counts ~ sex + age + education, data=data)

# collapse age to 3 levels
t2 <- collapse.table(t1, age=c("A", "A", "B", "B", "C", "C"))

# collapse age to 3 levels and pool education: "low" and "med" to "low"
t3 <- collapse.table(t1, age=c("A", "A", "B", "B", "C", "C"),
  education=c("low", "low", "high"))

# change labels for levels of education to 1:3
t4 <- collapse.table(t1, education=1:3)
```

Crossings

Crossings Interaction of Factors

Description

Given two ordered factors in a square, $n \times n$ frequency table, Crossings creates an $n-1$ column matrix corresponding to different degrees of difficulty in crossing from one level to the next, as described by Goodman (1972).

Usage

```
Crossings(...)
```

Arguments

```
...           Two factors
```

Value

For two factors of n levels, returns a binary indicator matrix of $n \times n$ rows and $n-1$ columns.

Author(s)

Michael Friendly and Heather Turner

References

Goodman, L. (1972). Some multiplicative models for the analysis of cross-classified data. In: *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, CA: University of California Press, pp. 649-696.

See Also

[glm](#), [gnm](#) for model fitting functions for frequency tables

[Diag](#), [Mult](#), [Symm](#), [Topo](#) for similar extensions to terms in model formulas.

Examples

```
data(Hauser79)
# display table
structable(~Father+Son, data=Hauser79)

hauser.indep <- gnm(Freq ~ Father + Son, data=Hauser79, family=poisson)
hauser.CR <- update(hauser.indep, ~ . + Crossings(Father,Son))
summarise(hauser.CR)

hauser.CRdiag <- update(hauser.indep, ~ . + Crossings(Father,Son) + Diag(Father,Son))
summarise(hauser.CRdiag)
```

 datasets

Information on Data Sets in Packages

Description

The `data` function is used both to load data sets from packages, and give a display of the names and titles of data sets in one or more packages, however it does not return a result that can be easily used to get additional information about the nature of data sets in packages.

The `datasets()` function is designed to produce a more useful summary display of data sets in one or more packages. It extracts the `class` and dimension information (`dim` or `codelength`) of each item, and formats these to provide additional descriptors.

Usage

```
datasets(package, allClass = FALSE, incPackage = length(package) > 1)
```

Arguments

<code>package</code>	a character vector giving the package(s) to look in
<code>allClass</code>	include all classes of the item (TRUE) or just the last class (FALSE)?
<code>incPackage</code>	include the package name in result?

Details

The requested packages must be installed, and are silently loaded in order to extract class and size information.

Value

A data.frame whose rows correspond to data sets found in package.

Author(s)

Michael Friendly, with R-help from Curt Seeliger

See Also

[data](#),

Examples

```
datasets("vcdExtra")
datasets(c("vcd", "vcdExtra"))
datasets("datasets")
```

DaytonSurvey

Dayton Student Survey on Substance Use

Description

This data, from Agresti (2002), Table 9.1, gives the result of a 1992 survey in Dayton Ohio of 2276 high school seniors on whether they had ever used alcohol, cigarettes and marijuana.

Usage

```
data(DaytonSurvey)
```

Format

A frequency data frame with 32 observations on the following 6 variables.

cigarette a factor with levels Yes No

alcohol a factor with levels Yes No

marijuana a factor with levels Yes No

sex a factor with levels female male

race a factor with levels white other

Freq a numeric vector

Details

Agresti uses the letters G (sex), R (race), A (alcohol), C (cigarette), M (marijuana) to refer to the table variables, and this usage is followed in the examples below.

Background variables include sex and race of the respondent (GR), typically treated as explanatory, so that any model for the full table should include the term `sex:race`. Models for the reduced table, collapsed over sex and race are not entirely unreasonable, but don't permit the estimation of the effects of these variables on the responses.

The full 5-way table contains a number of cells with counts of 0 or 1, as well as many cells with large counts, and even the ACM table collapsed over GR has some small cell counts. Consequently, residuals for these models in mosaic displays are best represented as standardized (adjusted) residuals.

Source

Agresti, A. (2002). *Categorical Data Analysis*, 2nd Ed., New York: Wiley-Interscience, Table 9.1, p. 362.

References

Thompson, L. (2009). *R (and S-PLUS) Manual to Accompany Agresti's Categorical Data*, <https://home.comcast.net/~lthompson221/Splusdiscrete2.pdf>

Examples

```
data(DaytonSurvey)

mod.GR <- glm(Freq ~ . + sex*race, data=DaytonSurvey, family=poisson) # mutual independence + GR
mod.homog.assoc <- glm(Freq ~ .^2, data=DaytonSurvey, family=poisson) # homogeneous association

# collapse over sex and race
Dayton.ACM <- aggregate(Freq ~ cigarette+alcohol+marijuana, data=DaytonSurvey, FUN=sum)
```

Detergent

Detergent preference data

Description

Cross-classification of a sample of 1008 consumers according to (a) the softness of the laundry water used, (b) previous use of detergent Brand M, (c) the temperature of laundry water used and (d) expressed preference for Brand X or Brand M in a blind trial.

Usage

```
data(Detergent)
```

Format

A 4-dimensional array resulting from cross-tabulating 4 variables for 1008 observations. The variable names and their levels are:

No	Name	Levels
1	Temperature	"High", "Low"
2	M_User	"Yes", "No"
3	Preference	"Brand X", "Brand M"
4	Water_softness	"Soft", "Medium", "Hard"

Source

Fienberg, S. E. (1980). *The Analysis of Cross-Classified Categorical Data* Cambridge, MA: MIT Press, p. 71.

References

Ries, P. N. & Smith, H. (1963). The use of chi-square for preference testing in multidimensional problems. *Chemical Engineering Progress*, 59, 39-43.

Examples

```
data(Detergent)

# example goes here
mosaic(Detergent, shade=TRUE)

(det.mod0 <- loglm(~ Preference + Temperature + M_User + Water_softness, data=Detergent))
# examine addition of two-way terms
add1(det.mod0, ~ .^2, test="Chisq")

# model for Preference as a response
(det.mod1 <- loglm(~ Preference + (Temperature * M_User * Water_softness), data=Detergent))
mosaic(det.mod0)
```

Description

Observational data on a sample of 1729 individuals, cross-classified in a 2⁵ table according to their sources of information (read newspapers, listen to the radio, do 'solid' reading, attend lectures) and whether they have good or poor knowledge regarding cancer. Knowledge of cancer is often treated as the response.

Usage

```
data(Dyke)
```

Format

A 5-dimensional array resulting from cross-tabulating 5 variables for 1729 observations. The variable names and their levels are:

No	Name	Levels
1	Knowledge	"Good", "Poor"
2	Reading	"No", "Yes"
3	Radio	"No", "Yes"
4	Lectures	"No", "Yes"
5	Newspaper	"No", "Yes"

Source

Fienberg, S. E. (1980). *The Analysis of Cross-Classified Categorical Data* Cambridge, MA: MIT Press, p. 85, Table 5-6.

References

Dyke, G. V. and Patterson, H. D. (1952). Analysis of factorial arrangements when the data are proportions. *Biometrics*, 8, 1-12.

Lindsey, J. K. (1993). *Models for Repeated Measurements* Oxford, UK: Oxford University Press, p. 57.

Examples

```
data(Dyke)

# independence model
mosaic(Dyke, shade=TRUE)

# baseline model, Knowledge as response
dyke.mod0 <- loglm(~ Knowledge + (Reading * Radio * Lectures * Newspaper), data=Dyke)
dyke.mod0

mosaic(dyke.mod0)
```

 expand.dft

Expand a frequency table to case form

Description

Converts a frequency table, given either as a table object or a data frame in frequency form to a data frame representing individual observations in the table.

Usage

```
expand.dft(x, var.names = NULL, freq = "Freq", ...)
```

```
expand.table(x, var.names = NULL, freq = "Freq", ...)
```

Arguments

x	A table object, or a data frame in frequency form containing factors and one numeric variable representing the cell frequency for that combination of factors.
var.names	A list of variable names for the factors, if you wish to override those already in the table
freq	The name of the frequency variable in the table
...	Other arguments passed down to <code>type.convert</code> . In particular, pay attention to <code>na.strings</code> (default: <code>na.strings=NA</code> if there are missing cells) and <code>as.is</code> (default: <code>as.is=FALSE</code> , converting character vectors to factors).

Details

`expand.table` is a synonym for `expand.dft`.

Value

A data frame containing the factors in the table and as many observations as are represented by the total of the `freq` variable.

Author(s)

Mark Schwarz

References

Posted on R-Help, Jan 20, 2009. <http://tolstoy.newcastle.edu.au/R/e6/help/09/01/1873.html>

See Also

[type.convert](#), [expandCategorical](#)

Examples

```
library(vcd)
art <- xtabs(~Treatment + Improved, data = Arthritis)
art
artdf <- expand.dft(art)
str(artdf)
```

 Fungicide

Carcinogenic Effects of a Fungicide

Description

Data from Gart (1971) on the carcinogenic effects of a certain fungicide in two strains of mice. Of interest is how the association between group (Control, Treated) and outcome (Tumor, No Tumor) varies with sex and strain of the mice.

Breslow (1976) used this data to illustrate the application of linear models to log odds ratios.

Usage

```
data(Fungicide)
```

Format

The data comprise a set of four 2 x 2 tables classifying 403 mice, either Control or Treated and whether or not a tumor was later observed. The four groups represent the combinations of sex and strain of mice. The format is: num [1:2, 1:2, 1:2, 1:2] 5 4 74 12 3 2 84 14 10 4 ... - attr(*, "dimnames")=List of 4 ..\$ group : chr [1:2] "Control" "Treated" ..\$ outcome: chr [1:2] "Tumor" "NoTumor" ..\$ sex : chr [1:2] "M" "F" ..\$ strain : chr [1:2] "1" "2"

Details

All tables have some small cells, so a continuity correction is recommended.

Source

Gart, J. J. (1971). The comparison of proportions: a review of significance tests, confidence intervals and adjustments for stratification. *International Statistical Review*, 39, 148-169.

References

Brewlow, N. (1976), Regression analysis of the log odds ratio: A method for retrospective studies, *Biometrics*, 32(3), 409-416.

Examples

```
data(Fungicide)
fung.lor <- loddsratio(Fungicide, correct=TRUE)
fung.lor
confint(fung.lor)

# visualize odds ratios in fourfold plots
cotabplot(Fungicide, panel=cotab_fourfold)
# -- fourfold() requires vcd >= 1.2-10
## Not run:
fourfold(Fungicide, p_adjust_method="none")
```

```
## End(Not run)
```

 Gilby

Clothing and Intelligence Rating of Children

Description

Schoolboys were classified according to their clothing and to their teachers rating of "dullness" (lack of intelligence), in a 5 x 7 table originally from Gilby (1911). Anscombe (1981) presents a slightly collapsed 4 x 6 table, used here, where the last two categories of clothing were pooled as were the first two categories of dullness due to small counts.

Both Dullnes and Clothing are ordered categories, so models and methods that examine their association in terms of ordinal categories are profitable.

Usage

```
data(Gilby)
```

Format

A 2-dimensional array resulting from cross-tabulating 2 variables for 1725 observations. The variable names and their levels are:

No	Name	Levels
1	Dullness	"Ment. defective", "Slow", "Slow Intell", "Fairly Intell", "Capable", "V.Able"
2	Clothing	"V.Well clad", "Well clad", "Passable", "Insufficient"

Source

Anscombe, F. J. (1981). *Computing in Statistical Science Through APL*. New York: Springer-Verlag, p. 302

References

Gilby, W. H. (1911). On the significance of the teacher's appreciation of general intelligence. *Biometrika*, 8, 93-108 (esp. p. 94). [Quoted by Kendall (1943,..., 1953) Table 13.1, p 320.]

Examples

```
data(Gilby)
```

```
mosaic(Gilby, shade=TRUE)
```

```
# correspondence analysis to see relations among categories
```

```

if(require(ca)){
  ca(Gilby)
  plot(ca(Gilby))
  title(xlab="Dimension 1", ylab="Dimension 2")
}

```

GKgamma

Calculate Goodman-Kruskal Gamma for ordered tables

Description

The Goodman-Kruskal γ statistic is a measure of association for ordinal factors in a two-way table proposed by Goodman and Kruskal (1954).

Usage

```
GKgamma(x, level = 0.95)
```

Arguments

x	A two-way frequency table, in matrix or table form. The rows and columns are considered to be ordinal factors
level	Confidence level for a significance test of $\gamma \neq 0$

Value

Returns an object of class "GKgamma" with 6 components, as follows

gamma	The gamma statistic
C	Total number of concordant pairs in the table
D	Total number of discordant pairs in the table
sigma	Standard error of gamma
CIlevel	Confidence level
CI	Confidence interval

Author(s)

Michael Friendly; original version by Laura Thompson

References

- Agresti, A. *Categorical Data Analysis*. John Wiley & Sons, 2002, pp. 57–59.
- Goodman, L. A., & Kruskal, W. H. (1954). Measures of association for cross classifications. *Journal of the American Statistical Association*, 49, 732-764.
- Goodman, L. A., & Kruskal, W. H. (1963). Measures of association for cross classifications III: Approximate sampling theory. *Journal of the American Statistical Association*, 58, 310-364.

See Also

[assocstats](#), [Kappa](#)

Examples

```
data(JobSat)
GKgamma(JobSat)
```

glmList	<i>Create a Model List Object</i>
---------	-----------------------------------

Description

`glmList` creates a `glmList` object containing a list of fitted `glm` objects with their names. `logglmList` does the same for `loglm` objects.

The intention is to provide object classes to facilitate model comparison, extraction, summary and plotting of model components, etc., perhaps using [lapply](#) or similar.

Usage

```
glmList(...)
logglmList(...)
```

Arguments

... One or more model objects, as appropriate to the function, optionally assigned names as in `list`.

Details

The arguments to `glmList` or `logglmList` are of the form `value` or `name=value`.

Any objects which do not inherit the appropriate class `glm` or `loglm` are excluded, with a warning.

Value

An object of class `glmList` `logglmList`, just like a `list`, except that each is given a name attribute.

Author(s)

Michael Friendly

See Also

The function `lolist` in package `Hmisc` is similar, but perplexingly more general.

The function `anova.glm` also handles `glm` objects

Examples

```
data(Mental)
indep <- glm(Freq ~ mental+ses,
             family = poisson, data = Mental)
Cscore <- as.numeric(Mental$ses)
Rscore <- as.numeric(Mental$mental)

coleff <- glm(Freq ~ mental + ses + Rscore:ses,
              family = poisson, data = Mental)
roweff <- glm(Freq ~ mental + ses + mental:Cscore,
              family = poisson, data = Mental)
linlin <- glm(Freq ~ mental + ses + Rscore:Cscore,
              family = poisson, data = Mental)

# use object names
mods <- glmlist(indep, coleff, roweff, linlin)
names(mods)

# assign new names
mods <- glmlist(Indep=indep, Col=coleff, Row=roweff, LinxLin=linlin)
names(mods)

summarise(mods)

#extract model components
unlist(lapply(mods, deviance))

res <- lapply(mods, residuals)
boxplot(as.data.frame(res), main="Residuals from various models")
```

Description

Data from the General Social Survey, 1991, on the relation between sex and party affiliation.

Usage

```
data(GSS)
```

Format

A data frame in frequency form with 6 observations on the following 3 variables.

sex a factor with levels female male
 party a factor with levels dem indep rep
 count a numeric vector

Source

Agresti, A. *Categorical Data Analysis* John Wiley & Sons, 2002, Table 3.11, p. 106.

Examples

```
data(GSS)
## maybe str(GSS) ; plot(GSS) ...
(GSStab <- xtabs(count ~ sex + party, data=GSS))

mod.glm <- glm(count ~ sex + party, family = poisson, data = GSS)
```

 Hauser79

Hauser (1979) Data on Social Mobility

Description

Hauser (1979) presented this two-way frequency table, cross-classifying occupational categories of sons and fathers in the United States.

Usage

```
data(Hauser79)
```

Format

A frequency data frame with 25 observations on the following 3 variables, representing the cross-classification of 19912 individuals by father's occupation and son's first occupation.

Son a factor with levels UpNM LoNM UpM LoM Farm
 Father a factor with levels UpNM LoNM UpM LoM Farm
 Freq a numeric vector

Source

R.M. Hauser (1979), Some exploratory methods for modeling mobility tables and other cross-classified data. In: K.F. Schuessler (Ed.), *Sociological Methodology*, 1980, Jossey-Bass, San Francisco, pp. 413-458.

References

Powers, D.A. and Xie, Y. (2008). *Statistical Methods for Categorical Data Analysis*, Bingley, UK: Emerald.

Examples

```

data(Hauser79)
str(Hauser79)

# display table
structable(~Father+Son, data=Hauser79)

#Examples from Powers & Xie, Table 4.15
# independence model
mosaic(Freq ~ Father + Son, data=Hauser79, shade=TRUE)

hauser.indep <- gnm(Freq ~ Father + Son, data=Hauser79, family=poisson)
mosaic(hauser.indep, ~Father+Son, main="Independence model", gp=shading_Friendly)

hauser.quasi <- update(hauser.indep, ~ . + Diag(Father,Son))
mosaic(hauser.quasi, ~Father+Son, main="Quasi-independence model", gp=shading_Friendly)

hauser.qsymm <- update(hauser.indep, ~ . + Diag(Father,Son) + Symm(Father,Son))
mosaic(hauser.qsymm, ~Father+Son, main="Quasi-symmetry model", gp=shading_Friendly)
#mosaic(hauser.qsymm, ~Father+Son, main="Quasi-symmetry model")

# numeric scores for row/column effects
Sscore <- as.numeric(Hauser79$Son)
Fscore <- as.numeric(Hauser79$Father)

# row effects model
hauser.roweff <- update(hauser.indep, ~ . + Father*Sscore)
summarise(hauser.roweff)

# uniform association
hauser.UA <- update(hauser.indep, ~ . + Fscore*Sscore)
summarise(hauser.UA)

# uniform association, omitting diagonals
hauser.UAdiag <- update(hauser.indep, ~ . + Fscore*Sscore + Diag(Father,Son))
summarise(hauser.UAdiag)

# Levels for Hauser 5-level model
levels <- matrix(c(
  2, 4, 5, 5, 5,
  3, 4, 5, 5, 5,
  5, 5, 5, 5, 5,
  5, 5, 5, 4, 4,
  5, 5, 5, 4, 1
), 5, 5, byrow=TRUE)

```

```

hauser.topo <- update(hauser.indep, ~ . + Topo(Son,Father, spec=levels))
mosaic(hauser.topo, ~Father+Son, main="Topological model", gp=shading_Friendly)

hauser.RC <- update(hauser.indep, ~ . + Mult(Son,Father))
mosaic(hauser.RC, ~Father+Son, main="RC model", gp=shading_Friendly)
summarise(hauser.RC)

# crossings models
hauser.CR <- update(hauser.indep, ~ . + Crossings(Father,Son))
mosaic(hauser.topo, ~Father+Son, main="Crossings model", gp=shading_Friendly)
summarise(hauser.CR)

hauser.CRdiag <- update(hauser.indep, ~ . + Crossings(Father,Son) + Diag(Father,Son))
summarise(hauser.CRdiag)

# compare model fit statistics
modlist <- glmlist(hauser.indep, hauser.roweff, hauser.UA, hauser.UAdiag, hauser.quasi, hauser.qsymm,
  hauser.topo, hauser.RC, hauser.CR, hauser.CRdiag)
sumry <- summarise(modlist)
sumry[order(sumry$AIC, decreasing=TRUE),]
# or, more simply
summarise(modlist, sortby="AIC")

op <- par(xpd=TRUE)
# mods <- gsub('hauser\\.',' ', rownames(sumry))
mods <- substring(rownames(sumry),8)
with(sumry,
{plot(Df, AIC, cex=1.3, pch=19, xlab='Degrees of freedom', ylab='AIC')
text(Df, AIC, mods, adj=c(0.5,-.5), col='red')
})
par(op)

```

Heart

Sex, Occupation and Heart Disease

Description

Classification of individuals by gender, occupational category and occurrence of heart disease

Usage

```
data(Heart)
```

Format

A 3-dimensional array resulting from cross-tabulating 3 variables for 21522 observations. The variable names and their levels are:

No	Name	Levels
1	Disease	"Disease", "None"
2	Gender	"Male", "Female"
3	Occup	"Unempl", "WhiteCol", "BlueCol"

Source

Karger, (1980).

Examples

```
data(Heart)

# example goes here
```

Heckman

Labour Force Participation of Married Women 1967-1971

Description

1583 married women were surveyed over the years 1967-1971, recording whether or not they were employed in the labor force.

The data, originally from Heckman & Willis (1977) provide an example of modeling longitudinal categorical data, e.g., with markov chain models for dependence over time.

Usage

```
data(Heckman)
```

Format

A 5-dimensional array resulting from cross-tabulating 5 variables for 1583 observations. The variable names and their levels are:

No	Name	Levels
1	e1971	"71Yes", "No"
2	e1970	"70Yes", "No"
3	e1969	"69Yes", "No"
4	e1968	"68Yes", "No"
5	e1967	"67Yes", "No"

Details

Lindsey (1993) fits an initial set of logistic regression models examining the dependence of employment in 1971 (e1971) on successive subsets of the previous years, e1970, e1969, ... e1967.

Alternatively, one can examine markov chain models of first-order (dependence on previous year), second-order (dependence on previous two years), etc.

Source

Lindsey, J. K. (1993). *Models for Repeated Measurements* Oxford, UK: Oxford University Press, p. 185.

References

Heckman, J.J. & Willis, R.J. (1977). "A beta-logistic model for the analysis of sequential labor force participation by married women." *Journal of Political Economy*, 85: 27-58

Examples

```
data(Heckman)

# independence model
mosaic(Heckman, shade=TRUE)
# same, as a loglm()
(heckman.mod0 <- loglm(~ e1971+e1970+e1969+e1968+e1967, data=Heckman))
mosaic(heckman.mod0, main="Independence model")

# first-order markov chain: bad fit
(heckman.mod1 <- loglm(~ e1971*e1970 + e1970*e1969 +e1969*e1968 + e1968*e1967, data=Heckman))
mosaic(heckman.mod1, main="1st order markov chain model")

# second-order markov chain: bad fit
(heckman.mod2 <- loglm(~ e1971*e1970*e1969 + e1970*e1969*e1968 +e1969*e1968*e1967, data=Heckman))
mosaic(heckman.mod2, main="2nd order markov chain model")

# second-order markov chain: fits OK
(heckman.mod3 <- loglm(~ e1971*e1970*e1969*e1968 + e1970*e1969*e1968*e1967, data=Heckman))
mosaic(heckman.mod2, main="3rd order markov chain model")
```

Hoyt

Minnesota High School Graduates

Description

Minnesota high school graduates of June 1930 were classified with respect to (a) Rank by thirds in their graduating class, (b) post-high school Status in April 1939 (4 levels), (c) Sex, (d) father's Occupational status (7 levels, from 1=High to 7=Low).

The data were first presented by Hoyt et al. (1959) and have been analyzed by Fienberg(1980), Plackett(1974) and others.

Usage

```
data(Hoyt)
```

Format

A 4-dimensional array resulting from cross-tabulating 4 variables for 13968 observations. The variable names and their levels are:

No	Name	Levels
1	Status	"College", "School", "Job", "Other"
2	Rank	"Low", "Middle", "High"
3	Occupation	"1", "2", "3", "4", "5", "6", "7"
4	Sex	"Male", "Female"

Details

Post high-school Status is natural to consider as the response. Rank and father's Occupation are ordinal variables.

Source

Fienberg, S. E. (1980). *The Analysis of Cross-Classified Categorical Data*. Cambridge, MA: MIT Press, p. 91-92.

R. L. Plackett, (1974). *The Analysis of Categorical Data*. London: Griffin.

References

Hoyt, C. J., Krishnaiah, P. R. and Torrance, E. P. (1959) Analysis of complex contingency tables, *Journal of Experimental Education* 27, 187-194.

See Also

[minn38](#) provides the same data as a data frame.

Examples

```
data(Hoyt)

# display the table
structable(Status+Sex ~ Rank+Occupation, data=Hoyt)

# mosaic for independence model
plot(Hoyt, shade=TRUE)

# examine all pairwise mosaics
pairs(Hoyt, shade=TRUE)

# collapse Status to College vs. Non-College
Hoyt1 <- collapse.table(Hoyt, Status=c("College", rep("Non-College",3)))
plot(Hoyt1, shade=TRUE)

#####
# fitting models with loglm, plotting with mosaic
```

```
#####

# fit baseline log-linear model for Status as response

hoyt.mod0 <- loglm(~ Status + (Sex*Rank*Occupation), data=Hoyt1)
hoyt.mod0
mosaic(hoyt.mod0, gp=shading_Friendly, main="Baseline model: Status + (Sex*Rank*Occ)")

# add one-way association of Status with factors
hoyt.mod1 <- loglm(~ Status * (Sex + Rank + Occupation) + (Sex*Rank*Occupation), data=Hoyt1)
hoyt.mod1
mosaic(hoyt.mod1, gp=shading_Friendly, main="Status * (Sex + Rank + Occ)")

# can we drop any terms?
drop1(hoyt.mod1, test="Chisq")

# assess model fit
anova(hoyt.mod0, hoyt.mod1)

# what terms to add?
add1(hoyt.mod1, ~.^2, test="Chisq")

# add interaction of Sex:Occupation on Status
hoyt.mod2 <- update(hoyt.mod1, ~.+Status:Sex:Occupation)
mosaic(hoyt.mod2, gp=shading_Friendly, main="Adding Status:Sex:Occupation")

# compare model fits
anova(hoyt.mod0, hoyt.mod1, hoyt.mod2)

# Alternatively, try stepwise analysis, heading toward the saturated model
steps <- step(hoyt.mod0, direction="forward", scope=~Status*Sex*Rank*Occupation)
# display anova
steps$anova
```

ICU

Death in the ICU

Description

The ICU data set consists of a sample of 200 subjects who were part of a much larger study on survival of patients following admission to an adult intensive care unit (ICU). The major goal of this study was to develop a logistic regression model to predict the probability of survival to hospital discharge of these patients and to study the risk factors associated with ICU mortality.

Usage

```
data(ICU)
```

Format

A data frame with 200 observations on the following 21 variables.

id Patient ID

died Died before discharge: a factor with levels No Yes

age Age: a numeric vector

sex Sex: a factor with levels Female Male

race Race: a factor with levels Black Other White

service Service at admission: a factor with levels Medical Surgery

cancer Cancer part of problem?: a factor with levels No Yes

renal History of chronic renal?: a factor with levels No Yes

infect Infection probable?: a factor with levels No Yes

cpr CPR prior to ICU admission?: a factor with levels No Yes

systolic Systolic blood pressure: a numeric vector

hrtrate Heart rate: a numeric vector

previcu Previous admit to ICU?: a factor with levels No Yes

admit Type of admission: a factor with levels Elective Emergency

fracture Fracture?: a factor with levels No Yes

po2 PO2 initial blood gas: a numeric vector

ph pH initial blood gas: a factor with levels <7.25 >=7.25

pco PCO2 initial blood gas: a factor with levels <=45 >45

bic Bicarbonate initial blood: a numeric vector

creatin Creatinine initial blood: a factor with levels <=2 >2

coma Consciousness at ICU: an ordered factor with levels None Stupor Coma

Details

Data were collected at Baystate Medical Center in Springfield, Massachusetts. The clinical aspects of this study are described in Lemeshow, Teres, Avrunin, and Pastides (1988).

Source

Hosmer and Lemeshow, *Applied Logistic Regression*, Wiley, (1989).

Lemeshow, S., Teres, D., Avrunin, J. S., Pastides, H. (1988). Predicting the Outcome of Intensive Care Unit Patients. *Journal of the American Statistical Association*, 83, 348-356.

References

Friendly, M. *Visualizing Categorical Data*, Cary, NC: SAS Institute, 2000, Appendix B.4.

Examples

```
data(ICU)
## maybe str(ICU) ; plot(ICU) ...
```

JobSat

Cross-classification of job satisfaction by income

Description

This data set is a contingency table of job satisfaction by income for a small sample of black males from the 1996 General Social Survey, as used by Agresti (2002) for an example.

Usage

```
data(JobSat)
```

Format

A 4 x 4 contingency table of income by satisfaction, with the following structure:

```
table [1:4, 1:4] 1 2 1 0 3 3 6 1 10 10 ...
- attr(*, "dimnames")=List of 2
..$ income      : chr [1:4] "< 15k" "15-25k" "25-40k" "> 40k"
..$ satisfaction: chr [1:4] "VeryD" "LittleD" "ModerateS" "VeryS"
```

Details

Both income and satisfaction are ordinal variables, and are so ordered in the table. Measures of association, visualizations, and models should take ordinality into account.

Source

Agresti, A. Categorical Data Analysis John Wiley & Sons, 2002, Table 2.8, p. 57.

Examples

```
data(JobSat)
assocstats(JobSat)
GKgamma(JobSat)
```

Kway

Fit All K-way Models in a GLM

Description

Generate and fit all 0-way, 1-way, 2-way, ... k-way terms in a glm.

This function is designed mainly for hierarchical loglinear models (or glms in the poisson family), where it is desired to find the highest-order terms necessary to achieve a satisfactory fit.

Using `anova` on the resulting `glm1ist` object will then give sequential tests of the pooled contributions of all terms of degree $k + 1$ over and above those of degree k .

This function is also intended as an example of a generating function for `glm1ist` objects, to facilitate model comparison, extraction, summary and plotting of model components, etc., perhaps using `lapply` or similar.

Usage

```
Kway(formula, family=poisson, data, ..., order = nt, prefix = "kway")
```

Arguments

<code>formula</code>	a two-sided formula for the 1-way effects in the model. The LHS should be the response, and the RHS should be the first-order terms connected by + signs.
<code>family</code>	a description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. (See <code>family</code> for details of family functions.)
<code>data</code>	an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>glm</code> is called.
<code>...</code>	Other arguments passed to <code>glm</code>
<code>order</code>	Highest order interaction of the models generated. Defaults to the number of terms in the model formula.
<code>prefix</code>	Prefix used to label the models fit in the <code>glm1ist</code> object.

Details

With `y` as the response in the formula, the 0-way (null) model is $y \sim 1$. The 1-way ("main effects") model is that specified in the `formula` argument. The k-way model is generated using the formula $. \sim .^k$. With the default `order = nt`, the final model is the saturated model.

As presently written, the function requires a two-sided formula with an explicit response on the LHS. For frequency data in table form (e.g., produced by `xtabs`) you the data argument is coerced to a `data.frame`, so you should supply the formula in the form `Freq ~ ...`.

Value

An object of class `glm1ist`, of length `order+1` containing the 0-way, 1-way, ... models up to degree `order`.

Author(s)

Michael Friendly and Heather Turner

See Also

[glmlist](#), [summarise](#)

Examples

```
## artificial data
factors <- expand.grid(A=factor(1:3), B=factor(1:2), C=factor(1:3), D=factor(1:2))
Freq <- rpois(nrow(factors), lambda=40)
df <- cbind(factors, Freq)

mods3 <- Kway(Freq ~ A + B + C, data=df, family=poisson)
summarise(mods3)
mods4 <- Kway(Freq ~ A + B + C + D, data=df, family=poisson)
summarise(mods4)

# JobSatisfaction data
data(JobSatisfaction, package="vcd")
modSat <- Kway(Freq ~ management+supervisor+own, data=JobSatisfaction, family=poisson, prefix="JSat")
summarise(modSat)
anova(modSat, test="Chisq")

# Rochdale data: very sparse, in table form
data(Rochdale, package="vcd")
## Not run:
modRoch <- Kway(Freq~EconActive + Age + HusbandEmployed + Child + Education + HusbandEducation + Asian + HouseholdW
  data=Rochdale, family=poisson)
summarise(modRoch)

## End(Not run)
```

loddsratio

Calculate Log Odds Ratios for Frequency Tables

Description

Computes (log) odds ratios and their asymptotic variance covariance matrix for (possibly) stratified data. Odds ratios are calculated for two array dimensions, separately for each level of all stratifying dimensions. This generalizes the [oddsratio](#) methods in the `vcd` package from 2 x 2 (x strata) tables to R x C (x strata) tables. In future versions, these functions may be renamed and or moved to the `vcd` package.

Usage

```
loddsratio(x, ...)
## Default S3 method:
loddsratio(x, strata = NULL, log = TRUE,
  ref = NULL, correct = any(x == 0), ...)
```

```
## S3 method for class 'loddsratio'
coef(object, log = object$log, ...)
## S3 method for class 'loddsratio'
vcov(object, log = object$log, ...)
## S3 method for class 'loddsratio'
print(x, log = x$log, ...)

as.array(x, ...)
## S3 method for class 'loddsratio'
as.array(x, log=x$log, ...)
```

Arguments

x	An object. For the default method a k-way matrix/table/array of frequencies. The number of margins has to be at least 2.
strata	Numeric or character indicating the margins of a k-way table x (with k greater than 2) that should be employed as strata. By default dimensions 3:length(dim(x)) are used.
ref	Numeric or character. Reference categories for the (non-stratum) row and column dimensions that should be employed for computing the odds ratios. By default, odds ratios for profile contrasts (or sequential contrasts, i.e., successive differences of adjacent categories) are used. See details below.
log	Logical. Should the results be displayed on a log scale or not? All internal computations are always on the log-scale but the results are transformed by default if log = TRUE.
correct	Logical. Should a continuity correction be applied (by adding 0.5 to all table entries) before computing odds ratios? By default, this not employed unless there are any zero cells in the table, but this correction is often recommended to reduce bias when some frequencies are small (Fleiss, 1981).
object	An object of class loddsratio as computed by loddsratio.
...	Arguments passed to methods.

Details

For an $R \times C$ table, (log) odds ratios are formed for the set of $(R-1) \times (C-1)$ 2×2 tables, corresponding to some set of contrasts among the row and column variables. The ref argument allows these to be specified in a general way.

ref = NULL (default) corresponds to “profile contrasts” (or sequential contrasts or successive differences) for ordered categories, i.e., R_1-R_2 , R_2-R_3 , R_3-R_4 , etc., and similarly for the column categories. These are sometimes called “local odds ratios”.

ref=1 gives contrasts with the first category; ref=dim(x) gives contrasts with the last category; ref = c(2, 4) or ref = list(2, 4) corresponds to the reference being the second category in rows and the fourth in columns.

Combinations like ref = list(NULL, 3) are also possible, as are character vectors, e.g., ref = c("foo", "bar") also works ("foo" pertaining again to the row reference and "bar" to column reference).

Note that all such parameterizations are equivalent, in that one can derive all other possible odds ratios from any non-redundant set.

Value

An object of class `loddsratio`, with the following components:

<code>coefficients</code>	A named vector, of length $(R-1) \times (C-1) \times \text{prod}(\text{dim}(x)[\text{strata}])$ containing the log odds ratios. Use the <code>coef</code> method to extract these from the object, and the <code>confint</code> method for confidence intervals. For a two-way table, the names for the log odds ratios are constructed in the form <code>Ri:Rj/Ci:Cj</code> using the table names for rows and columns. For a stratified table, the names are constructed in the form <code>Ri:Rj/Ci:CjLk</code> .
<code>vcov</code>	Variance covariance matrix of the log odds ratios.
<code>dimnames</code>	Dimension names for the log odds ratios, considered as a table of size $c(R-1, C-1, \text{dim}(x)[\text{strata}])$. Use the <code>dim</code> and <code>dimnames</code> methods to extract these and manipulate the log odds ratios in relation to the original table.
<code>dim</code>	Corresponding dimension vector.
<code>contrasts</code>	A matrix <code>C</code> , such that <code>C %% as.vector(log(x))</code> gives the log odds ratios. Each row corresponds to one log odds ratio, and is all zero, except for 4 elements of <code>c(1, -1, -1, 1)</code> for a given 2×2 subtable.
<code>log</code>	A logical, indicating whether the value of <code>log</code> in the original call.

Note

The method of calculation is an example of the use of the delta method described by Agresti (1990), Section 12.1.7, giving estimates of log odds ratios and their asymptotic covariance matrix.

The `coef` method returns the `coefficients` component as a vector of length $(R-1) \times (C-1) \times \text{prod}(\text{dim}(x)[\text{strata}])$. The `dim` and `dimnames` methods provide the proper attributes for treating the `coefficients` vector as an $(R-1) \times (C-1) \times \text{strata}$ array. `as.matrix` and `as.array` methods are also provided for this purpose.

The `confint` method computes confidence intervals for the log odds ratios (or for odds ratios, with `log=FALSE`). The `coeftest` method prints the asymptotic standard errors, z tests (standardized log odds ratios), and the corresponding p values.

Author(s)

Achim Zeileis and Michael Friendly

References

- A. Agresti (1990), *Categorical Data Analysis*. New York: Wiley.
 Fleiss, J. L. (1981). *Statistical Methods for Rates and Proportions*. 2nd Edition. New York: Wiley.
 M. Friendly (2000), *Visualizing Categorical Data*. SAS Institute, Cary, NC.

See Also

[oddsratio](#), [confint](#),
[coeftest](#) for z-tests of significance

Examples

```

## artificial example
set.seed(1)
x <- matrix(rpois(5 * 3, 7), ncol = 5, nrow = 3)
dimnames(x) <- list(Row=head(letters, 3), Col=tail(letters, 5))

x_lor <- loddsratio(x)
coef(x_lor)
x_lor
confint(x_lor)
if(require("lmtest")) coeftest(x_lor)

## 2 x 2 x k cases
#data(CoalMiners, package="vcd")
lor.CM <- loddsratio(CoalMiners)
lor.CM
coef(lor.CM)
confint(lor.CM)
confint(lor.CM, log=FALSE)

# odds ratio plot
lor.CM.df <- as.data.frame(lor.CM)
lor.CM.df <- within(lor.CM.df,
{lower<-LOR-ASE
  upper<-LOR+ASE}
)
range <- c(min(lor.CM.df$lower), max(lor.CM.df$upper))
with(lor.CM.df, {
plot(LOR ~ as.numeric(Age), type='b', pch=16, xaxt='n',
  ylim=range,
xlab="Age", ylab="Log odds ratio: Wheeze x Breathlessness",
main="CoalMiners data: Log odds ratio plot")
axis(side=1, at=as.numeric(Age), labels=Age)
segments(as.numeric(Age), lower, as.numeric(Age), upper)
}
)
# fit linear models using WLS
abline(lm(LOR ~ as.numeric(Age), weights=1/ASE^2, data=lor.CM.df), col="blue")
age <- seq(25, 60, by = 5)
qmod <- lm(LOR ~ poly(age,2), weights=1/ASE^2, data=lor.CM.df)
lines(fitted(qmod), col = "red", lwd=2)

## 2 x k x 2
lor.Emp <-loddsratio(Employment)
lor.Emp
confint(lor.Emp)

# visualize the log odds ratios
lor.Emp.a <- as.array(lor.Emp)
matplot(lor.Emp.a, type='b', xaxt='n', pch=15:16, cex=1.5,

```

```

ylab='log odds ratio: Employment Status x Length',
xlab='Employment Length',
xlim=c(0.2, 5),
main="Employment status data")
abline(h=0, col='gray')
axis(side=1, at=1:5, labels=rownames(lor.Emp.a))
text(0.3, lor.Emp.a[1,], colnames(lor.Emp.a), pos=4, col=1:2)
text(0.5, max(lor.Emp.a[1,])+.1, "Layoff cause")

## R x C case
#data(Hauser79, package="vcdExtra")
hauser.tab <- xtabs(Freq ~ Father+Son, data=Hauser79)
(lor.hauser <- loddsratio(hauser.tab))
confint(lor.hauser)

# odds ratio plot
op <- par(xpd=TRUE)
matplot(as.matrix(lor.hauser), type='b', ylab='log odds ratio',
xlab="Son's status comparisons", xaxt='n', xlim=c(1,4.5), ylim=c(-.5,3),
main="Hauser79 data")
abline(h=0, col='gray')
axis(side=1, at=1:4, labels=colnames(lor.hauser))
text(4, as.matrix(lor.hauser)[4,], rownames(lor.hauser), pos=4, col=1:4)
text(4, 3, "Father's status")
par(op)

## 4 way tables
data(Punishment, package="vcd")
punish <- xtabs(Freq ~ memory + attitude + age + education, data = Punishment)
mosaic(~ age + education + memory + attitude, data = punish, keep = FALSE,
gp = gpar(fill = grey.colors(2)), spacing = spacing_highlighting,
rep = c(attitude = FALSE))

lor.pun <- loddsratio(punish)
lor.pun
confint(lor.pun)
if(require("lmtest")) coefTest(lor.pun)

# visualize the log odds ratios, by education
lor.pun.a <- as.array(lor.pun)
matplot(lor.pun.a, type='b', xaxt='n', pch=15:17, cex=1.5,
ylab='log odds ratio: Attitude x Memory',
xlab='Age', xlim=c(0.5, 3), ylim=c(-2, 1),
main="Attitudes toward corporal punishment")
abline(h=0, col='gray')
axis(side=1, at=1:3, labels=rownames(lor.pun.a))
text(0.5, lor.pun.a[1,], colnames(lor.pun.a), pos=4, col=1:3)
text(0.6, max(lor.pun.a[1,])+.2, "Education")

# visualize the log odds ratios, by age
matplot(t(lor.pun.a), type='b', xaxt='n', pch=15:17, cex=1.5,

```

```

ylab='log odds ratio: Attitude x Memory',
xlab='Education', xlim=c(0.5, 3), ylim=c(-2, 1),
main="Attitudes toward corporal punishment")
abline(h=0, col='gray')
axis(side=1, at=1:3, labels=colnames(lor.pun.a))
text(0.5, lor.pun.a[,1], rownames(lor.pun.a), pos=4, col=1:3)
text(0.6, max(lor.pun.a[,1])+.2, "Age")

# fit linear model using WLS
lor.pun.df <- as.data.frame(lor.pun)
pun.mod1 <- lm(LOR ~ as.numeric(age) * as.numeric(education), data=lor.pun.df, weights=1/ASE^2)
anova(pun.mod1)

```

Mammograms

Mammogram Ratings

Description

Kundel & Polansky (2003) give (possibly contrived) data on a set of 110 mammograms rated by two readers.

Usage

```
data(Mammograms)
```

Format

A frequency table in matrix form. The format is: num [1:4, 1:4] 34 6 2 0 10 8 5 1 2 8 ... -
attr(*, "dimnames")=List of 2 ..\$ Reader2: chr [1:4] "Absent" "Minimal" "Moderate" "Severe" ..\$
Reader1: chr [1:4] "Absent" "Minimal" "Moderate" "Severe"

Source

Kundel, H. L. & Polansky, M. (2003), "Measurement of Observer Agreement", *Radiology*, **228**, 303-308, Table A1

Examples

```

data(Mammograms)
B <- agreementplot(Mammograms, main="Mammogram ratings")
# agreement measures
B
Kappa(Mammograms)

## other displays
mosaic(Mammograms, shade=TRUE)

sieve(Mammograms, pop = FALSE, shade = TRUE)
labeling_cells(text = Mammograms, gp_text = gpar(fontface = 2, cex=1.75))(as.table(Mammograms))

```

meanResiduals	<i>Average Residuals within Factor Levels</i>
---------------	---

Description

Computes the mean working residual from a model fitted using Iterative Weighted Least Squares for each level of a factor or interaction of factors.

Usage

```
meanResiduals(object, by, standardized=TRUE, as.table=TRUE, ...)
```

Arguments

object	model object for which <code>object\$residuals</code> gives the working residuals and <code>object\$weights</code> gives the working weights.
by	a list of factors.
standardized	logical: if TRUE, the mean residuals are standardized to be approximately standard normal.
as.table	logical: if TRUE the result is returned as a table cross-classified by the factors passed to <code>by</code> .
...	currently ignored

Value

If `as.table == TRUE`, the mean residuals cross-classified by the factors passed to `by`, otherwise a vector of mean residuals. In either case the returned object has a single attribute, "weights" which gives the weight associated with each grouped residual.

Author(s)

Heather Turner

Examples

```
data(yaish)
## Fit a conditional independence model, leaving out
## the uninformative subtable for dest == 7:
CImodel <- gnm(Freq ~ educ*orig + educ*dest, family = poisson,
              data = yaish, subset = (dest != 7))

## compute mean residuals over origin and destination
meanResiduals(CImodel, model.frame(CImodel)[c("orig", "dest")])

## display mean residuals for origin and destination
mosaic(CImodel, ~orig+dest)
```

```
## non-aggregated residuals
res1 <- meanResiduals(CImodel,
                      model.frame(CImodel)[c("educ", "orig", "dest")])

res2 <- residuals(CImodel, type = "pearson")

all.equal(res1[, , ], res2[, , ])
```

Mental

Mental impariment and parents SES

Description

A 6 x 4 contingency table representing the cross-classification of mental health status (mental) of 1660 young New York residents by their parents' socioeconomic status (ses).

Usage

```
data(Mental)
```

Format

A data frame frequency table with 24 observations on the following 3 variables.

ses an ordered factor with levels 1 < 2 < 3 < 4 < 5 < 6

mental an ordered factor with levels Well < Mild < Moderate < Impaired

Freq cell frequency: a numeric vector

Details

Both ses and mental can be treated as ordered factors or integer scores. For ses, 1="High" and 6="Low".

Source

Haberman, S. J. *The Analysis of Qualitative Data: New Developments*, Academic Press, 1979, Vol. II, p. 375.

Srole, L.; Langner, T. S.; Michael, S. T.; Kirkpatrick, P.; Opler, M. K. & Rennie, T. A. C. *Mental Health in the Metropolis: The Midtown Manhattan Study*, NYU Press, 1978, p. 289

References

Friendly, M. *Visualizing Categorical Data*, Cary, NC: SAS Institute, 2000, Appendix B.7.

Examples

```

data(Mental)
str(Mental)
(Mental.tab <- xtabs(Freq ~ ses+mental, data=Mental))

# mosaic and sieve plots
mosaic(Mental.tab, gp=shading_Friendly)
sieve(Mental.tab, gp=shading_Friendly)

library(ca)
plot(ca(Mental.tab), main="Mental impairment & SES")
title(xlab="Dim 1", ylab="Dim 2")

```

Mobility

Social Mobility data

Description

Data on social mobility, recording the occupational category of fathers and their sons.

Usage

```
data(Mobility)
```

Format

A 2-dimensional array resulting from cross-tabulating 2 variables for 19912 observations. The variable names and their levels are:

```

No  Name
1  Son's_Occupation}\tab \code{"UpNonMan", "LoNonMan", "UpManual", "LoManual", "Farm"}\cr 2\tab \code{

```

Source

Falguerolles, A. de and Mathieu, J. R. (1988). *Proceedings of COMPSTAT 88*, Copenhagen, Denmark, Springer-Verlag.

Featherman, D. L. and Hauser, R. M. Occupations and social mobility in the United States. *Sociological Microjournal*, 12, Fiche 62. Copenhagen: Sociological Institute.

Examples

```

data(Mobility)

# example goes here

```

`modFit`*Brief Summary of Model Fit for a glm or loglm Object*

Description

Formats a brief summary of model fit for a `glm` or `loglm` object, showing the likelihood ratio `Chisq` (`df`) value and or `AIC`. Useful for inclusion in a plot title or annotation.

Usage

```
modFit(x, ...)
## S3 method for class 'glm'
modFit(x, stats="chisq", digits=2, ...)
## S3 method for class 'loglm'
modFit(x, stats="chisq", digits=2, ...)
```

Arguments

<code>x</code>	A <code>glm</code> or <code>loglm</code> object
<code>...</code>	Arguments passed down
<code>stats</code>	One or more of <code>chisq</code> or <code>aic</code> , determining the statistics displayed.
<code>digits</code>	Number of digits after the decimal point in displayed statistics.

Value

A character string containing the formatted values of the chosen statistics.

Author(s)

Michael Friendly

See Also

[summarise](#)

Examples

```
data(Mental)
(Mental.tab <- xtabs(Freq ~ ses+mental, data=Mental))
(Mental.mod <- loglm(~ses+mental, Mental.tab))
Mental.mod
modFit(Mental.mod)

# use to label mosaic()
mosaic(Mental.mod, main=paste("Independence model,", modFit(Mental.mod)))
```

mosaic.glm	<i>Mosaic plots for fitted generalized linear and generalized nonlinear models</i>
------------	--

Description

Produces mosaic plots (and other plots in the [strucplot](#) framework) for a log-linear model fitted with [glm](#) or for a generalized nonlinear model fitted with [gnm](#).

These methods extend the range of strucplot visualizations well beyond the models that can be fit with [loglm](#). They are intended for models for counts using the Poisson family (or quasi-poisson), but should be sensible as long as (a) the response variable is non-negative and (b) the predictors visualized in the strucplot are discrete factors.

Usage

```
## S3 method for class 'glm'
mosaic(x, formula = NULL, panel = mosaic, type = c("observed", "expected"), residuals = NULL,
       residuals_type = c("pearson", "deviance", "rstandard"), gp = shading_hcl, gp_args = list(), ...)
## S3 method for class 'glm'
sieve(x, ...)
## S3 method for class 'glm'
assoc(x, ...)
```

Arguments

x	A glm or gnm object. The response variable, typically a cell frequency, should be non-negative.
formula	A one-sided formula with the indexing factors of the plot separated by '+'. A formula must be provided unless <code>x\$data</code> inherits from class "table" – in which case the indexing factors of this table are used, or the factors in <code>x\$data</code> (or <code>model.frame(x)</code> if <code>x\$data</code> is an environment) exactly cross-classify the data – in which case this set of cross-classifying factors are used.
panel	Panel function used to draw the plot for visualizing the observed values, residuals and expected values. Currently, one of "mosaic", "assoc", or "sieve" in vcd .
type	A character string indicating whether the "observed" or the "expected" values of the table should be visualized by the area of the tiles or bars.
residuals	An optional array or vector of residuals corresponding to the cells in the data, for example, as calculated by <code>residuals.glm(x)</code> , <code>residuals.gnm(x)</code> .
residuals_type	If the <code>residuals</code> argument is NULL, residuals are calculated internally and used in the display. In this case, <code>residuals_type</code> can be "pearson", "deviance" or "rstandard". Otherwise (when <code>residuals</code> is supplied), <code>residuals_type</code> is used as a label for the legend in the plot.
gp	Object of class "gpar", shading function or a corresponding generating function (see strucplot Details and shadings). Ignored if <code>shade = FALSE</code> .

`gp_args` A list of arguments for the shading-generating function, if specified.
 ... Other arguments passed to the `panel` function e.g., `mosaic`

Details

For both poisson family generalized linear models and loglinear models, standardized residuals provided by `rstandard` (sometimes called adjusted residuals) are often preferred because they have constant unit asymptotic variance.

The `sieve` and `assoc` methods are simple convenience interfaces to this plot method, setting the `panel` argument accordingly.

Value

The structable visualized by `strucplot` is returned invisibly.

Author(s)

Heather Turner, Michael Friendly, with help from Achim Zeileis

See Also

[glm](#), [gnm](#), [plot.loglm](#), [mosaic](#)

Examples

```
GSStab <- xtabs(count ~ sex + party, data=GSS)
# using the data in table form
mod.glm1 <- glm(Freq ~ sex + party, family = poisson, data = GSStab)
res <- residuals(mod.glm1)
std <- rstandard(mod.glm1)

# For mosaic.default(), need to re-shape residuals to conform to data
stdtab <- array(std, dim=dim(GSStab), dimnames=dimnames(GSStab))
mosaic(GSStab, gp=shading_Friendly, residuals=stdtab, residuals_type="Std\nresiduals",
       labeling = labeling_residuals)

# Using externally calculated residuals with the glm() object
mosaic.glm(mod.glm1, residuals=std, labeling = labeling_residuals, shade=TRUE)

# Using residuals_type
mosaic.glm(mod.glm1, residuals_type="rstandard", labeling = labeling_residuals, shade=TRUE)

## Ordinal factors and structured associations
data(Mental)
xtabs(Freq ~ mental+ses, data=Mental)
long.labels <- list(set_varnames = c(mental="Mental Health Status", ses="Parent SES"))

# fit independence model
# Residual deviance: 47.418 on 15 degrees of freedom
indep <- glm(Freq ~ mental+ses,
```

```

        family = poisson, data = Mental)

long.labels <- list(set_varnames = c(mental="Mental Health Status",
                                   ses="Parent SES"))
mosaic(indep,residuals_type="rstandard", labeling_args = long.labels, labeling=labeling_residuals)
# or, show as a sieve diagram
mosaic(indep, labeling_args = long.labels, panel=sieve, gp=shading_Friendly)

# fit linear x linear (uniform) association. Use integer scores for rows/cols
Cscore <- as.numeric(Mental$ses)
Rscore <- as.numeric(Mental$mental)

linlin <- glm(Freq ~ mental + ses + Rscore:Cscore,
             family = poisson, data = Mental)
mosaic(linlin,residuals_type="rstandard",
       labeling_args = long.labels, labeling=labeling_residuals, suppress=1, gp=shading_Friendly,
       main="Lin x Lin model")

## Goodman Row-Column association model fits even better (deviance 3.57, df 8)
if (require(gnm)) {
  Mental$mental <- C(Mental$mental, treatment)
  Mental$ses <- C(Mental$ses, treatment)
  RC1model <- gnm(Freq ~ ses + mental + Mult(ses, mental),
                 family = poisson, data = Mental)

  mosaic(RC1model,residuals_type="rstandard",
        labeling_args = long.labels, labeling=labeling_residuals, suppress=1, gp=shading_Friendly,
        main="RC1 model")
}

##### UCB Admissions data, fit using glm()

structable(Dept ~ Admit+Gender,UCBAdmissions)

berkeley <- as.data.frame(UCBAdmissions)
berk.glm1 <- glm(Freq ~ Dept * (Gender+Admit), data=berkeley, family="poisson")
summary(berk.glm1)
mosaic(berk.glm1, gp=shading_Friendly, labeling=labeling_residuals, formula=~Admit+Dept+Gender)
# the same, displaying studentized residuals; note use of formula to reorder factors in the mosaic
mosaic(berk.glm1, residuals_type="rstandard", labeling=labeling_residuals, shade=TRUE,
       formula=~Admit+Dept+Gender, main="Model: [DeptGender][DeptAdmit]")

## all two-way model
berk.glm2 <- glm(Freq ~ (Dept + Gender + Admit)^2, data=berkeley, family="poisson")
summary(berk.glm2)
mosaic.glm(berk.glm2, residuals_type="rstandard", labeling = labeling_residuals, shade=TRUE,
          formula=~Admit+Dept+Gender, main="Model: [DeptGender][DeptAdmit][AdmitGender]")
anova(berk.glm1, berk.glm2, test="Chisq")

# Add 1 df term for association of [GenderAdmit] only in Dept A
berkeley <- within(berkeley, dept1AG <- (Dept=='A')*(Gender=='Female')*(Admit=='Admitted'))
berkeley[1:6,]
berk.glm3 <- glm(Freq ~ Dept * (Gender+Admit) + dept1AG, data=berkeley, family="poisson")

```

```
summary(berk.glm3)
mosaic.glm(berk.glm3, residuals_type="rstandard", labeling = labeling_residuals, shade=TRUE,
formula=~Admit+Dept+Gender, main="Model: [DeptGender][DeptAdmit] + DeptA*[GA]")
anova(berk.glm1, berk.glm3, test="Chisq")
```

mosaic3d

3D Mosaic Plots

Description

Produces a 3D mosaic plot for a contingency table (or a `link[MASS]{loglm}` model) using the [rgl-package](#).

Generalizing the 2D mosaic plot, this begins with a given 3D shape (a unit cube), and successively sub-divides it along the X, Y, Z dimensions according to the table margins, generating a nested set of 3D tiles. The volume of the resulting tiles is therefore proportional to the frequency represented in the table cells. Residuals from a given loglinear model are then used to color or shade each of the tiles.

This is a developing implementation. The arguments and details are subject to change.

Usage

```
mosaic3d(x, ...)

## S3 method for class 'loglm'
mosaic3d(x, type = c("observed", "expected"), residuals_type = c("pearson", "deviance"), ...)

## Default S3 method:
mosaic3d(x, expected = NULL, residuals = NULL,
type = c("observed", "expected"), residuals_type = NULL,
shape = cube3d(alpha = alpha), alpha = 0.5,
spacing = 0.1, split_dir = 1:3, shading = shading_basic, zero_size=.05,
label_edge,
labeling_args = list(), newpage = TRUE, box=FALSE, ...)
```

Arguments

x	A <code>link[MASS]{loglm}</code> model object. Alternatively, a multidimensional array or table or structable of frequencies in a contingency table. In the present implementation, the dimensions are taken in sequential order. Use <code>link[base]{aperm}</code> or structable to change this.
expected	optionally, for contingency tables, an array of expected frequencies of the same dimension as x, or alternatively the corresponding loglinear model specification as used by <code>link[stats]{loglin}</code> or <code>link[MASS]{loglm}</code> (see structable for details).
residuals	optionally, an array of residuals of the same dimension as x (see details).

type	a character string indicating whether the "observed" or the "expected" frequencies in the table should be visualized by the volume of the 3D tiles.
residuals_type	a character string indicating the type of residuals to be computed when none are supplied. If residuals is NULL, residuals_type must be one of "pearson" (default; giving components of Pearson's chi-squared), "deviance" (giving components of the likelihood ratio chi-squared), or "FT" for the Freeman-Tukey residuals. The value of this argument can be abbreviated.
shape	The initial 3D shape on which the mosaic is based. Typically this is a call to an rgl function, and must produce a shape3d object. The default is a "unit cube" on (-1, +1), with transparency specified by alpha.
alpha	Specifies the transparency of the 3D tiles used to compose the 3D mosaic.
spacing	A number or vector giving the total amount of space used to separate the 3D tiles along each of the dimensions of the table. The values specified are re-cycled to the number of table dimensions.
split_dir	A numeric vector composed of the integers 1:3 or a character vector composed of c("x", "y", "z"), where split_dir[i] specifies the axis along which the tiles should be split for dimension i of the table. The values specified are re-cycled to the number of table dimensions.
shading	A function, taking an array or vector of residuals for the given model, returning a vector of colors. At present, only the default shading=shading_basic is provided. This is roughly equivalent to the use of the shade argument in mosaicplot or to the use of gp=shading_Friendly in mosaic .
zero_size	The radius of a small sphere used to mark zero cells in the display.
label_edge	A character vector composed of c("-", "+") indicating whether the labels for a given table dimension are to be written at the minima ("-") or maxima ("+") of the <i>other</i> dimensions in the plot. The default is rep(c('-', '+'), each=3, length=ndim), meaning that the first three table variables are labeled at the minima, and successive ones at the maxima.
labeling_args	This argument is intended to be used to specify details of the rendering of labels for the table dimensions, but at present has no effect.
newpage	logical indicating whether a new page should be created for the plot or not.
box	logical indicating whether a bounding box should be drawn around the plot.
...	Other arguments passed down to mosaic.default or 3D functions.

Details

Friendly (1995), Friendly [Sect. 4.5](2000) and Theus and Lauer (1999) have all used the idea of 3D mosaic displays to explain various aspects of loglinear models (the iterative proportional fitting algorithm, the structure of various models for 3-way and n-way tables, etc.), but no implementation of 3D mosaics was previously available.

For the default method, residuals, used to color and shade the 3D tiles, can be passed explicitly, or, more typically, are computed as needed from observed and expected frequencies. In this case, the expected frequencies are optionally computed for a specified loglinear model given by the expected argument. For the loglm method, residuals and observed frequencies are calculated from the model object.

Value

Invisibly, the list of shape3d objects used to draw the 3D mosaic, with names corresponding to the concatenation of the level labels, separated by ":".

Author(s)

Michael Friendly, with the help of Duncan Murdoch and Achim Zeileis

References

Friendly, M. (1995). Conceptual and Visual Models for Categorical Data, *The American Statistician*, **49**, 153-160.

Friendly, M. *Visualizing Categorical Data*, Cary NC: SAS Institute, 2000. Web materials: <http://www.datavis.ca/books/vcd/>.

Theus, M. & Lauer, S. R. W. (1999) Visualizing Loglinear Models. *Journal of Computational and Graphical Statistics*, **8**, 396-412.

See Also

[strucplot](#), [mosaic](#), [mosaicplot](#)

[loglin](#), [loglm](#) for details on fitting loglinear models

Examples

```
# 2 x 2 x 2
mosaic3d(Bartlett, box=TRUE)
# compare with expected frequencies under model of mutual independence
mosaic3d(Bartlett, type="expected", box=TRUE)

# 2 x 2 x 3
mosaic3d(Heart, box=TRUE)

## Not run:
# 2 x 2 x 2 x 3
# illustrates a 4D table
mosaic3d(Detergent)

# compare 2D and 3D mosaics
demo("mosaic-hec")

## End(Not run)
```

print.Kappa	<i>Print Kappa</i>
-------------	--------------------

Description

This is a replacement for the print.Kappa method in vcd, adding display of z values to the vcd version.

Usage

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

Arguments

x	A Kappa object
...	Other arguments

Value

Returns the Kappa object, invisibly.

Author(s)

Michael Friendly

See Also

[confint.Kappa](#)

Examples

```
data("SexualFun")  
Kappa(SexualFun)
```

split3d	<i>Subdivide a 3D Object</i>
---------	------------------------------

Description

Subdivides a shape3d object or a list of shape3d objects into objects of the same shape along a given dimension according to the proportions or frequencies specified in vector(s).

split3d is the basic workhorse used in [mosaic3d](#), but may be useful in other contexts.

range3d and center3d are utility functions, also useful in other contexts.

Usage

```

split3d(obj, ...)

## S3 method for class 'shape3d'
split3d(obj, p, dim, space = 0.1, ...)

## S3 method for class 'list'
split3d(obj, p, dim, space = 0.1, ...)

range3d(obj)

center3d(obj)

```

Arguments

obj	A shape3d object, or a list composed of them
...	Other arguments for split3d methods
p	For a single shade3d object, a vector of proportions (or a vector of non-negative numbers which will be normed to proportions) indicating the number of subdivisions and their scaling along dimension dim. For a list of shade3d objects, a matrix whose columns indicate the subdivisions of each object.
dim	The dimension along which the object is to be subdivided. Either an integer: 1, 2, or 3, or a character: "x", "y", or "z".
space	The total space used to separate the copies of the object along dimension dim. The unit inter-object space is therefore $\text{space}/(\text{length}(p)-1)$.

Details

The resulting list of shape3d objects is actually composed of *copies* of the input object(s), scaled according to the proportions in p and then translated to make their range along the splitting dimension equal to that of the input object(s).

Value

split3d returns a list of shape3d objects.

range3d returns a 2 x 3 matrix, whose first row contains the minima on dimensions x, y, z, and whose second row contains the maxima.

center3d returns a numeric vector containing the means of the minima and maxima on dimensions x, y, z.

Author(s)

Duncan Murdoch, with refinements by Michael Friendly

See Also[mosaic3d](#)[shapelist3d](#) for the plotting of lists of shape3d objects.**Examples**

```

if (require(rgl)) {
  open3d()
  cube <- cube3d(alpha=0.4)
  sl1 <- split3d(cube, c(.2, .3, .5), 1)
  col <- c("#FF000080", "#E5E5E580", "#0000FF80")
  shapelist3d(sl1, col=col)

  open3d()
  p <- matrix(c(.6, .4, .5, .5, .2, .8), nrow=2)
  sl2 <- split3d(sl1, p, 2)
  shapelist3d(sl2, col=col)
}

```

summarise

*Brief Summary of Model Fit for glm and loglm Models***Description**

For glm objects, the print and summary methods give too much information if all one wants to see is a brief summary of model goodness of fit, and there is no easy way to display a compact comparison of model goodness of fit for a collection of models fit to the same data.

All loglm models have equivalent glm forms, but the print and summary methods give quite different results

summarise provides a brief summary for one or more glm or loglm models This implementation is experimental, and is subject to change.

Usage

```

summarise(object, ...)

## S3 method for class 'glm'
summarise(object, ..., test = NULL)
## S3 method for class 'glmmlist'
summarise(object, ..., test = NULL, sortby=NULL)

## S3 method for class 'loglm'
summarise(object, ...)
## S3 method for class 'loglmmlist'
summarise(object, ..., sortby=NULL)

```

Arguments

object, ...	objects of class <code>glm</code> , typically the result of a call to <code>glm</code> , or a list of objects for the <code>glm</code> method. Alternatively, objects of class <code>loglm</code> or a "loglm" object
test	Not used in the current implementation.
sortby	For <code>glm</code> and <code>loglm</code> objects, either a numeric or character string specifying the column in the result for which the rows are sorted (in decreasing order).

Value

A data frame (also of class `anova`) with columns `c("LR Chisq", "Df", "Pr(>Chisq)", "AIC", "BIC")`. Row names are taken from the names of the model object(s).

Author(s)

Michael Friendly

See Also

[glm](#), [loglm](#), [modFit](#)

Examples

```
data(Mental)
indep <- glm(Freq ~ mental+ses,
            family = poisson, data = Mental)
summarise(indep)
Cscore <- as.numeric(Mental$ses)
Rscore <- as.numeric(Mental$mental)

coleff <- glm(Freq ~ mental + ses + Rscore:ses,
            family = poisson, data = Mental)
roweff <- glm(Freq ~ mental + ses + mental:Cscore,
            family = poisson, data = Mental)
linlin <- glm(Freq ~ mental + ses + Rscore:Cscore,
            family = poisson, data = Mental)

# make a glm
mods <- glm(indep, coleff, roweff, linlin)
summarise(mods)
```

TV

TV Viewing Data

Description

This data set TV comprises a 5 x 11 x 3 contingency table based on audience viewing data from Nielsen Media Research for the week starting November 6, 1995.

Usage

```
data(TV)
```

Format

A 5 x 11 x 3 array of cell frequencies with the following structure:

```
int [1:5, 1:11, 1:3] 146 244 233 174 294 151 181 161 183 281 ...
- attr(*, "dimnames")=List of 3
..$ Day      : chr [1:5] "Monday" "Tuesday" "Wednesday" "Thursday" ...
..$ Time     : chr [1:11] "8:00" "8:15" "8:30" "8:45" ...
..$ Network: chr [1:3] "ABC" "CBS" "NBC"
```

Details

The original data, `tv.dat`, contains two additional networks: "Fox" and "Other", with small frequencies. These levels were removed in the current version. There is also a fourth factor, transition State transition (turn the television Off, Switch channels, or Persist in viewing the current channel). The TV data here includes only the Persist observations.

Source

The original data, `tv.dat`, came from the initial implementation of mosaic displays in R by Jay Emerson. Similar data had been used by Hartigan and Kleiner (1984) as an illustration.

References

Hartigan, J. A. & Kleiner, B. A Mosaic of Television Ratings *The American Statistician*, 1984, 38, 32-35

Examples

```
data(TV)
structable(TV)
doubledecker(TV)

# reduce number of levels of Time
TV.df <- as.data.frame.table(TV)
levels(TV.df$Time) <- rep(c("8:00-8:59", "9:00-9:59", "10:00-10:44"), c(4, 4, 3))
TV2 <- xtabs(Freq ~ Day + Time + Network, TV.df)

# re-label for mosaic display
levels(TV.df$Time) <- c("8", "9", "10")
# fit mode of joint independence, showing association of Network with Day*Time
mosaic(~ Day + Network + Time, data = TV.df, expected = ~ Day:Time + Network, legend = FALSE)
# with doubledecker arrangement
mosaic(~ Day + Network + Time, data = TV.df, expected = ~ Day:Time + Network,
       split = c(TRUE, TRUE, FALSE), spacing = spacing_highlighting, legend = FALSE)
```

Vietnam

Student Opinion About the War in Vietnam

Description

A survey of student opinion on the Vietnam War was taken at the University of North Carolina at Chapel Hill in May 1967 and published in the student newspaper. Students were asked to fill in ballot papers stating which policy out of A,B,C or D they supported. Responses were cross-classified by gender/year.

The result is a 2 x 5 x 4 contingency table, with factors sex, yr and resp.

Usage

```
data(Vietnam)
```

Format

A data frame in frequency form with 40 observations on the following 4 variables.

sex a factor with levels F M

yr an ordered factor with levels Fresh < Soph < Jr < Sr < Grad

resp a factor with levels A B C D

Freq a numeric vector

Details

The response categories can be considered ordered in terms of attitude about the war in Vietnam:

A defeat North Vietnam by widespread bombing and land invasion

B follow the present policy

C withdraw troops to strong points and open negotiations on elections involving the Viet Cong

D immediate withdrawal of all U.S. troops

Source

Aitkin, M. A., Anderson, D., Francis, B., & Hinde, J. *Statistical Modelling in GLIM*, Clarendon Press, 1989

References

Friendly, M. *Visualizing Categorical Data*, Cary, NC: SAS Institute, 2000, Appendix B.12.

Examples

```
data(Vietnam)
str(Vietnam)
# make it into a table
Vietnam.table <- xtabs(Freq ~ sex+yr+resp, data=Vietnam)
structable(Vietnam.table)
```

Yamaguchi87

Occupational Mobility in Three Countries

Description

Yamaguchi (1987) presented this three-way frequency table, cross-classifying occupational categories of sons and fathers in the United States, United Kingdom and Japan. This data set has become a classic for models comparing two-way mobility tables across layers corresponding to countries, groups or time (e.g., Goodman and Hout, 1998; Xie, 1992).

The US data were derived from the 1973 OCG-II survey; those for the UK from the 1972 Oxford Social Mobility Survey; those for Japan came from the 1975 Social Stratification and Mobility survey. They pertain to men aged 20-64.

Usage

```
data(Yamaguchi87)
```

Format

A frequency data frame with 75 observations on the following 4 variables. The total sample size is 28887.

Son a factor with levels UpNM LoNM UpM LoM Farm

Father a factor with levels UpNM LoNM UpM LoM Farm

Country a factor with levels US UK Japan

Freq a numeric vector

Details

Five status categories – upper and lower nonmanuals (UpNM, LoNM), upper and lower manuals (UpM, LoM), and Farm) are used for both fathers' occupations and sons' occupations.

Upper nonmanuals are professionals, managers, and officials; lower nonmanuals are proprietors, sales workers, and clerical workers; upper manuals are skilled workers; lower manuals are semi-skilled and unskilled nonfarm workers; and farm workers are farmers and farm laborers.

Some of the models from Xie (1992), Table 1, are fit in `demo(yamaguchi-xie)`.

Source

Yamaguchi, K. (1987). Models for comparing mobility tables: toward parsimony and substance, *American Sociological Review*, vol. 52 (Aug.), 482-494, Table 1

References

- Goodman, L. A. and Hout, M. (1998). Statistical Methods and Graphical Displays for Analyzing How the Association Between Two Qualitative Variables Differs Among Countries, Among Groups, Or Over Time: A Modified Regression-Type Approach. *Sociological Methodology*, 28 (1), 175-230.
- Xie, Yu (1992). The log-multiplicative layer effect model for comparing mobility tables. *American Sociological Review*, 57 (June), 380-395.

Examples

```

data(Yamaguchi87)
# reproduce Table 1
structable(~ Father + Son + Country, Yamaguchi87)
# create table form
Yama.tab <- xtabs(Freq ~ Son + Father + Country, data=Yamaguchi87)

# define mosaic labeling_args for convenient reuse in 3-way displays
largs <- list(rot_labels=c(right=0), offset_varnames = c(right = 0.6), offset_labels = c(right = 0.2),
             set_varnames = c(Son="Son's status", Father="Father's status")
            )

#####
# Fit some models & display mosaics

# Mutual independence
yama.indep <- glm(Freq ~ Son + Father + Country, data=Yamaguchi87, family=poisson)
anova(yama.indep)

mosaic(yama.indep, ~Son+Father, main="[S][F] ignoring country")
mosaic(yama.indep, ~Country + Son + Father, condvars="Country",
       labeling_args=largs,
       main='[S][F][C] Mutual independence')

# no association between S and F given country ('perfect mobility')
# asserts same associations for all countries
yama.noRC <- glm(Freq ~ (Son + Father) * Country, data=Yamaguchi87, family=poisson)
anova(yama.noRC)
mosaic(yama.noRC, ~~Country + Son + Father, condvars="Country",
       labeling_args=largs,
       main="[SC][FC] No [SF] (perfect mobility)")

# ignore diagonal cells
yama.quasi <- update(yama.noRC, ~ . + Diag(Son,Father):Country)
anova(yama.quasi)
mosaic(yama.quasi, ~Son+Father, main="Quasi [S][F]")

## see also:
# demo(yamaguchi-xie)
##

```

Index

- *Topic **array**
 - expand.dft, 16
 - *Topic **attribute**
 - collapse.table, 10
 - *Topic **category**
 - GKgamma, 20
 - loddsratio, 33
 - print.Kappa, 49
 - *Topic **datasets**
 - Abortion, 4
 - Alligator, 5
 - Bartlett, 7
 - Caesar, 8
 - Cancer, 9
 - DaytonSurvey, 13
 - Detergent, 14
 - Dyke, 15
 - Fungicide, 18
 - Gilby, 19
 - GSS, 22
 - Hauser79, 23
 - Heart, 25
 - Heckman, 26
 - Hoyt, 27
 - ICU, 29
 - JobSat, 31
 - Mammograms, 38
 - Mental, 40
 - Mobility, 41
 - TV, 52
 - Vietnam, 54
 - Yamaguchi87, 55
 - *Topic **data**
 - datasets, 12
 - *Topic **dplot**
 - split3d, 49
 - *Topic **hplot**
 - mosaic.glm, 43
 - mosaic3d, 46
 - *Topic **htest**
 - GKgamma, 20
 - print.Kappa, 49
 - *Topic **manip**
 - collapse.table, 10
 - Crossings, 11
 - expand.dft, 16
 - *Topic **models**
 - Crossings, 11
 - glm1ist, 21
 - Kway, 31
 - meanResiduals, 39
 - modFit, 42
 - mosaic.glm, 43
 - summarise, 51
 - *Topic **multivariate**
 - mosaic.glm, 43
 - *Topic **nonlinear**
 - meanResiduals, 39
 - *Topic **package**
 - datasets, 12
 - vcdExtra-package, 3
 - *Topic **regression**
 - meanResiduals, 39
 - *Topic **utilities**
 - glm1ist, 21
 - modFit, 42
-
- Abortion, 4
 - Alligator, 5
 - anova, 32
 - anova.glm, 22
 - as.array(loddsratio), 33
 - as.data.frame, 32
 - as.data.frame.loddsratio(loddsratio), 33
 - as.matrix.loddsratio(loddsratio), 33
 - assoc.glm(mosaic.glm), 43
 - assocstats, 21

- Bartlett, 7
- Caesar, 8
- caesar, 8
- Cancer, 9
- center3d (split3d), 49
- coef.loddsratio (loddsratio), 33
- coefstest, 35
- collapse.table, 10
- confint, 35
- confint.Kappa, 49
- Crossings, 3, 11
- data, 12, 13
- datasets, 12
- DaytonSurvey, 13
- Detergent, 14
- Diag, 3, 12
- dim.loddsratio (loddsratio), 33
- dimnames.loddsratio (loddsratio), 33
- Dyke, 15
- expand.dft, 10, 16
- expand.table (expand.dft), 16
- expandCategorical, 17
- family, 32
- Fungicide, 18
- Gilby, 19
- GKgamma, 20
- glm, 12, 43, 44
- glm1ist, 3, 21, 32, 33, 52
- gnm, 12, 43, 44
- GSS, 22
- Hauser79, 23
- Heart, 25
- Heckman, 26
- Hoyt, 27
- ICU, 29
- JobSat, 31
- Kappa, 21
- Kway, 3, 31
- lapply, 21
- l1ist, 22
- loddsratio, 33
- loglin, 48
- loglm, 43, 48
- loglm1ist, 52
- loglm1ist (glm1ist), 21
- Mammograms, 38
- margin.table, 10
- meanResiduals, 39
- Mental, 40
- minn38, 28
- Mobility, 41
- modFit, 42, 52
- mosaic, 4, 44, 47, 48
- mosaic.glm, 3, 43
- mosaic.loglm, 3
- mosaic3d, 3, 46, 49, 51
- mosaicplot, 47, 48
- Mult, 12
- oddsratio, 33, 35
- plot.loglm, 44
- print.GKgamma (GKgamma), 20
- print.Kappa, 49
- print.loddsratio (loddsratio), 33
- range3d (split3d), 49
- shadings, 43
- shapelist3d, 51
- sieve.glm (mosaic.glm), 43
- split3d, 49
- strucplot, 43, 44, 48
- structable, 46
- summarise, 3, 33, 42, 51
- Symm, 3, 12
- Topo, 3, 12
- TV, 52
- type.convert, 17
- vcdExtra (vcdExtra-package), 3
- vcdExtra-package, 3
- vcov.loddsratio (loddsratio), 33
- Vietnam, 54
- Yamaguchi87, 55