

Bergsma, Wicher and van der Ark, Andries
Categorical marginal models: quite extensive package for the estimation of marginal models for categorical data

Monograph (Manual)

Original citation:

Bergsma, Wicher and van der Ark, Andries (2015) Categorical marginal models: quite extensive package for the estimation of marginal models for categorical data. 0.8. Comprehensive R Archive Network.

Originally available from <http://cran.r-project.org/>

This version available at: <http://eprints.lse.ac.uk/61309/>

Available in LSE Research Online: March 2015

© 2015 The Authors

LSE has developed LSE Research Online so that users may access research output of the School. Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Users may download and/or print one copy of any article(s) in LSE Research Online to facilitate their private study or for non-commercial research. You may not engage in further distribution of the material or use it for any profit-making activities or any commercial gain. You may freely distribute the URL (<http://eprints.lse.ac.uk>) of the LSE Research Online website.

Package ‘cmm’

February 19, 2015

Type Package

Title Categorical Marginal Models

Version 0.8

Date 2015-01-14

Author Wicher Bergsma and Andries van der Ark

Maintainer L. A. van der Ark <L.A.vanderArk@uva.nl>

Description Quite extensive package for the estimation of marginal models for categorical data.

License GPL (>= 2)

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2015-01-15 07:39:46

R topics documented:

cmm-package	2
Antisemitism	3
BodySatisfaction	4
ClarenceThomas	6
ConstraintMatrix	8
DesignMatrix	10
DirectSum	13
DutchConcern	14
DutchPolitics	15
ErieCounty	17
EVS	18
GSS93	19
JoinModels	21
LaborParticipation	23
MarginalMatrix	24
MarginalModelFit	26
MarihuanaAlcohol	30

ModelStatistics	32
NES	34
NKPS	35
SampleStatistics	37
Smoking	39
SpecifyCoefficient	40
TestCronbachAlpha	42

Index	48
--------------	-----------

cmm-package	<i>Categorical Marginal Models</i>
-------------	------------------------------------

Description

A package accompanying the book *Marginal Models for Dependent, Clustered, and Longitudinal Categorical Data* by Bergsma, Croon, & Hagenaars, 2009. Its purpose is fitting and testing of marginal models.

Details

Package: cmm
 Type: Package
 Version: 2.0
 Date: 2009-04-07
 License: GPL (>= 2)

The package contains principal functions for analyzing marginal models for categorical data. All functions are illustrated using examples from the book *Marginal Models for Dependent, Clustered, and Longitudinal Categorical Data* (Bergsma, Croon, & Hagenaars, 2009).

The package contains the following functions [ConstraintMatrix](#) [DesignMatrix](#) [DirectSum](#) [JoinModels](#) [MarginalMatrix](#) [MarginalModelFit](#) [ModelStatistics](#) [SampleStatistics](#) [SpecifyCoefficient](#)

The package contains the following data sets [Antisemitism](#) [BodySatisfaction](#) [ClarenceThomas](#) [DutchConcern](#) [DutchPolitics](#) [ErieCounty](#) [EVS](#) [GSS93](#) [LaborParticipation](#) [MarihuanaAlcohol](#) [NES](#) [NKPS](#) [NKPS2](#) [Smoking](#)

Thanks are due to Irena Mikolajun for reporting bugs;

Author(s)

Wicher P. Bergsma L. Andries van der Ark Maintainer: Wicher P. Bergsma <w.p.bergsma@lse.ac.uk>.

References

Bergsma, W. P. (1997). *Marginal models for categorical data*. Tilburg, The Netherlands: Tilburg University Press. http://stats.lse.ac.uk/bergsma/pdf/bergsma_phdthesis.pdf

Bergsma, W. P., Croon, M. A., & Hagenaaars, J. A. P. (2009). Marginal models for dependent, clustered, and longitudinal categorical data. Berlin: Springer.

Antisemitism

Change in antisemitism after seeing a movie

Description

A classical data set that has been used several times in the past, but not analyzed by means of the methods advocated in this book (Glock, 1955; Campbell & Clayton, 1961; Hagenaaars, 1990, pp. 215-233, and Hagenaaars, 1990, Section 5.3). The data are from a panel study among 503 white Christians living in and around Baltimore. The study's purpose was to determine the effect of seeing the film 'Gentleman's Agreement' on reducing the level of antisemitism (Glock, 1955, p. 243). Antisemitism was measured in November 1947 (variable *A*) prior to the movie being locally shown and consisted of three categories : 1 = high, 2 = moderate, and 3 = low. Antisemitism was measured again in May 1948 (variable *B*). In addition, the respondents were asked whether or not they had (voluntary) seen the movie, which had been shown in Baltimore theaters during the period between the two interviews (variable *X*). The experimental group (with $X = 1$) consisted of those respondents who saw the movie; the control group (with $X = 2$) consisted of those who did not. The data are tabulated in Bergsma, Croon, and Hagenaaars (2009, Table 5.8).

Section 5.2.2 in Bergsma, Croon, and Hagenaaars (2009).

Usage

data(GSS93)

Format

A data frame with 496 observations on the following three variables.

X Seen the film (factor): 1 = Seen; 2 = Not seen;

A Antisemitism at Time 1 (ordered): 1 = High; 2 = Moderate; 3 = Low.

B Antisemitism at Time 2 (ordered): 1 = High; 2 = Moderate; 3 = Low.

Source

Glock (1955).

References

Bergsma, W. P., Croon, M. A., & Hagenaaars, J. A. P. (2009). Marginal models for dependent, clustered, and longitudinal categorical data. Berlin: Springer.

Campbell & Clayton (1961)

Glock (1955)

Hagenaaars, 1990

Examples

```

data(Antisemitism)

## Sample marginal distributions
# at applied to data gives vectorized 2x2x3 table TXR (Time x Seen film or not x Response)
at <- MarginalMatrix(c("X","A","B"), list(c("X","A"), c("X","B")), c(2,3,3));
stats = SampleStatistics(
  dat = Antisemitism,
  coeff = at,
  Labels = c("T","X","R"),
  CoefficientDimensions = c(2,2,3))

## Models for table XR given T

# at1 applied to data gives vectorized conditional 2x3 table XR (XR conditional on T<-1)
at1 <- MarginalMatrix(c("X", "A", "B"), list(c("X", "A")), c(2, 3, 3));
# at2 applied to data gives vectorized conditional 2x3 table XR (XR conditional on T<-2)
at2 <- MarginalMatrix(c("X", "A", "B"), list(c("X", "B")), c(2, 3, 3));

bt1 <- ConstraintMatrix(c("X", "R"), list(c("X"), c("R")), c(2, 3));
bt2 <- ConstraintMatrix(c("X", "R"), list(c("X"), c("R")), c(2, 3));

model1 <- list(bt1, "log", at1);
model2 <- list(bt2, "log", at2);

# model1 doesn't converge, I don't know the reason and am trying to find out
# (it does converge in the Mathematica programme).
fit = MarginalModelFit(
  dat = Antisemitism,
  model = model2,
  Labels = c("X","R"),
  CoefficientDimensions = c(2,3),
  MaxSteps=100,
  ShowProgress=10,
  MaxStepSize=.5)

```

BodySatisfaction

Body satisfaction for seven body parts

Description

A group of 301 university students (204 women and 97 men) answered questions about their degrees of satisfaction with different parts or aspects of their body by completing the Body Esteem Scale (Franzoi & Shields, 1984; Bekker, Croon, & Vermaas, 2002). This scale consisted of 22 items (not counting the items concerning gender-specific body parts), seven of which will be considered here. These seven items loaded highest on the first unrotated principal component, with loadings higher than .70. Principal component analysis was used to discover whether the separate expressions of satisfaction with the different body aspects can be seen as just an expression of the general underlying satisfaction with the body as a whole or whether more underlying dimensions are needed

(for the interested reader: two rotated factors were needed to explain the correlations among all the 22 items, one having to do with the general appearance of the body and the other with the satisfaction with the parts of one's face; the items chosen here all belong to the first factor.) The data are tabulated in Bergsma, Croon, and Hagenaaars (2009, Table 2.5, Table 2.7).

See Sections 2.2.2 and 3.1 of Bergsma, Croon, and Hagenaaars (2009).

Several worked examples involving this data set are listed below but more can be found at <http://stats.lse.ac.uk/bergsma/cmm/Rfiles/BodySatisfaction.R>"

Usage

```
data(BodySatisfaction)
```

Format

A data frame with 301 observations on the following variables.

Gender (factor): 0 = Male; 1 = Female.

Thighs (ordered): 1 = Very dissatisfied; 2 = Moderately dissatisfied; 3 = Slightly satisfied. 4 = Moderately satisfied. 5 = Very satisfied.

BodyBuild (ordered): see Thighs

Buttocks (ordered): see Thighs

Hips (ordered): see Thighs

Legs (ordered): see Thighs

Figure (ordered): see Thighs

Weight (ordered): see Thighs

Source

Bekker, Croon, & Vermaas (2002).

References

Bekker, M.H.J., Croon, M.A., & Vermaas, S. (2002). Inner body and outward appearance- the relationship between orientation toward outward appearance, body awareness and symptom perception. *Personality and Individual Differences*, 33, 213-225.

Bergsma, W. P., Croon, M. A., & Hagenaaars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. New York: Springer.

Franzoi, S.L. & Shields, S.A. (1984). The Body-Esteem Scale: Multidimensional structure and sex differences in a college population. *Journal of Personality Assessment*, 48, 173-178.

Examples

```
data(BodySatisfaction)
```

```
## Reproduction of Table 2.5 in Bergsma, Croon, & Hagenaaars (2009)
sapply(2:8,function(i){table(BodySatisfaction[,i])})
```

```

## Table 2.6 in Bergsma, Croon and Hagnaars (2009).
## Loglinear parameters for marginal table IS
## We provide two to obtain the parameters

## Reproduction of Table 2.7 in Bergsma, Croon, & Hagnaars (2009)
Table.2.7.male <-
  sapply(2:8,function(i){table(BodySatisfaction[BodySatisfaction[1]=="Male",i])})
Table.2.7.male
#totals
apply(Table.2.7.male,2,sum)
#means
apply(Table.2.7.male,2,function(x){sum(c(1:5)*x/sum(x))})
#standard deviations
sqrt(apply(Table.2.7.male,2,function(x){(sum(c(1:5)^2*x/sum(x))-(sum(c(1:5)*x/sum(x))^2))})

## Not run:
dat <- BodySatisfaction[,2:8]      # omit first column corresponding to gender

# matrix producing 1-way marginals, ie the 7x5 table IS
at75 <- MarginalMatrix(var = c(1, 2, 3, 4, 5, 6, 7),
  marg = list(c(1),c(2),c(3),c(4),c(5),c(6),c(7)),
  dim = c(5, 5, 5, 5, 5, 5, 5))
# First method: the "coefficients" are the log-probabilities,
# from which all the (loglinear) parameters are calculated
stats <- SampleStatistics(dat = dat,
  coeff = list("log", at75),
  CoefficientDimensions=c(7, 5),
  Labels=c("I", "S"),
  ShowCoefficients = FALSE,
  ShowParameters = TRUE)

# Second method: the "coefficients" are explicitly specified as being the
# (highest-order) loglinear parameters
loglinpar75 <- SpecifyCoefficient("LoglinearParameters", c(7,5) )
stats = SampleStatistics(dat = dat,
  coeff = list(loglinpar75, at75),
  CoefficientDimensions = c(7, 5),
  Labels = c("I", "S"))

## End(Not run)
#For further worked examples from the book,
# see http://stats.lse.ac.uk/bergsma/cmm/R files/BodySatisfaction.R

```

Description

Clarence Thomas was nominated in 1991 as member of the U.S. Supreme Court by President George H. W. Bush. The nomination provoked some public debate because of Clarence Thomas' race (black) and because of his allegedly extremely conservative social and political views. A panel of U.S. citizens was interviewed regarding their opinion on Clarence Thomas' candidacy during September 3-5 (A) and on October 9 (B). After the first wave, more precisely on September 25, a charge of sexual harassment was brought against Clarence Thomas by his former aide, Anita Hill. Source: CBS News and New York Times 2001.

The data are tabulated in Bergsma, Croon, and Hagenaaars (2009, Table 5.6) and were also used in Bergsma & Croon (2005).

Section 5.2.1 in Bergsma, Croon, and Hagenaaars (2009).

Usage

```
data(ClarenceThomas)
```

Format

A data frame with 991 observations on the following variables.

A Opinion on Clarence Thomas during first wave, Sept 3-5, 1991 (factor): 1 = Favorable; 2 = Unfavorable; 3 = Undecided; 4 = Haven't heard enough;

B Opinion on Clarence Thomas during second wave, Oct 9, 1991 (factor): 1 = Favorable; 2 = Unfavorable; 3 = Undecided; 4 = Haven't heard enough;

Source

CBS News and New York Times 2001.

References

Bergsma, W. P., Croon, M. A., & Hagenaaars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. Berlin: Springer

Bergsma, W. P., & Croon, M. A. (2005). Analyzing categorical data by marginal models. In L. A. van der Ark, M. A. Croon, & K. Sijtsma (Eds.), *New developments in categorical data analysis for the social and behavioral sciences*. Mahwah, NJ: Erlbaum.

Examples

```
data(ClarenceThomas)
```

```
#####  
## Marginal homogeneity: 01=02
```

```
# at24 produces vectorized 2x4 table TR (Time x Response)  
at24 <- MarginalMatrix(c("A", "B"), list(c("A"), c("B")), c(4, 4));
```

```
# marginal homogeneity  
bt1 <- ConstraintMatrix(c("T", "R"), list(c("T"), c("R")), c(2, 4));
```



```

model1 <- list(bt1, "log", at24);

# only first two categories are equated
bt2 <- rbind(
  c(1, 0, 0, 0, -1, 0, 0, 0),
  c(0, 1, 0, 0, 0, -1, 0, 0));
model2 <- c(bt2, "log", at24);

pi11 <- MarginalModelFit(ClarenceThomas, model1,
  MaxSteps = 500,
  ShowProgress = 20,
  MaxStepSize = .5,
  CoefficientDimensions = c(2, 4),
  Labels = c("T", "R"),
  Title = "Clarence Thomas data, MH");

#####
## Marginal homogeneity: P1=P2
# P1 and P2 are O1 and O2 conditioned on not being in category 4

# at24 produces vectorized 2x4 table TR (Time x Response)
at24 <- MarginalMatrix(c("A", "B"), list(c("A"), c("B")), c(4, 4));

# specify conditional probabilities using generalized exp-log notation
at1 <- rbind(c(1, 0, 0, 0), c(0, 1, 0, 0), c(0, 0, 1, 0), c(1, 1, 1, 0));
at1 <- DirectSum(at1, at1);
at2 <- rbind(c(1, 0, 0, -1), c(0, 1, 0, -1), c(0, 0, 1, -1));
at2 <- DirectSum(at2, at2);
coeff <- list(list(diag(6), at2, at1), list("exp", "log", "identity"));

# marginal homogeneity
bt1 <- ConstraintMatrix(c("T", "R"), list(c("T"), c("R")), c(2, 3));

model1 <- list(bt1, coeff, at24);

pi11 <- MarginalModelFit(ClarenceThomas, model1,
  MaxSteps = 500,
  ShowProgress = 20,
  MaxStepSize = .5,
  CoefficientDimensions = c(2, 3),
  Labels = c("T", "R"),
  Title = "Clarence Thomas data, MH");

```

ConstraintMatrix

ConstraintMatrix

Description

Returns hierarchical model constraint matrix, i.e., nullspace of design matrix

Usage

```
ConstraintMatrix(var, suffconfigs, dim, SubsetCoding = "Automatic")
```

Arguments

var	character or numeric vector containing variables
suffconfigs	subvector or list of subvectors of var indicating the sufficient configurations or highest order interactions in model
dim	numeric vector indicating the dimension of var (must be same length as var)
SubsetCoding	allows a (character) type or a matrix to be assigned to variables for each element of suffconfigs, see examples

Details

The model $\mu_{ij} = \alpha + \beta_i + \gamma_j$ has parametric form and can equivalently be described using constraints on the μ_{ij} , by $\mu_{ij} - \mu_{il} - \mu_{kj} + \mu_{kl} = 0$. Returns the transpose of the null space of `DesignMatrix(var, marg, dim)`. Rows normally sum to zero. See [DesignMatrix](#) for more details.

Value

matrix

Author(s)

W. P. Bergsma <w.p.bergsma@lse.ac.uk>

References

Bergsma, W. P. (1997). *Marginal models for categorical data*. Tilburg, The Netherlands: Tilburg University Press. http://stats.lse.ac.uk/bergsma/pdf/bergsma_phdthesis.pdf

Bergsma, W. P., Croon, M. A., & Hagenaaars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. Berlin: Springer.

See Also

[ConstraintMatrix](#), [DesignMatrix](#), [DirectSum](#), [MarginalMatrix](#)

Examples

```
# Constraint matrix for independence model
var <- c("A", "B")
suffconfigs <- list(c("A"), c("B"))
dim <- c(3, 3)
ConstraintMatrix(var, suffconfigs, dim)
# notation in one line
ConstraintMatrix(c("A", "B"), list(c("A"), c("B")), c(3, 3))
```

```

# Constraint matrix for saturated model, two short specifications giving same result
ConstraintMatrix(c("A","B"),c("A","B"),c(3,3))
ConstraintMatrix(c("A","B"),list(c("A","B")),c(3,3))

# Constraint matrix for univariate quadratic regression model
var <- c("A")
suffconfigs <- c("A")
dim <- c(5)
ConstraintMatrix(var,suffconfigs,dim,SubsetCoding=list(c("A"),"Quadratic"))
# notation in one line
ConstraintMatrix(c("A"),c("A"),c(5),SubsetCoding=list(c("A"),"Quadratic"))

# Constraint matrix for linear by nominal model, various methods:
# simplest method which assumes equidistant centered scores:
ConstraintMatrix(
  var = c("A", "B"),
  suffconfigs = c("A", "B"),
  dim = c(3, 3),
  SubsetCoding = list(c("A", "B"), list("Linear", "Nominal")))
# alternative specification with same result as above:
ConstraintMatrix(
  var = c("A", "B"),
  suffconfigs = c("A", "B"),
  dim = c(3, 3),
  SubsetCoding = list(c("A", "B"), list(rbind(c(-1, 0, 1)), rbind(c(1, 0, 0), c(0, 1, 0)))))
# specifying your own category scores
scores <- c(1,2,5);
ConstraintMatrix(
  var = c("A", "B"),
  suffconfigs = c("A", "B"),
  dim = c(3, 3),
  SubsetCoding = list(c("A", "B"), list(rbind(scores), "Nominal")))

# Constraint matrix for nominal by nominal model, equating parameters of
# last two categories of second variable:
ConstraintMatrix(var = c("A", "B"), suffconfigs = c("A", "B"), dim = c(3,3),
  SubsetCoding = list(c("A", "B"), list("Nominal", rbind(c(1, 0, 0), c(0, 1, 1)))))

```

DesignMatrix

DesignMatrix

Description

Returns hierarchical model design matrix

Usage

```
DesignMatrix(var, suffconfigs, dim, SubsetCoding = "Automatic", MakeSubsets=TRUE)
```

Arguments

var	character or numeric vector containing variables
suffconfigs	subvector or list of subvectors of var indicating the sufficient configurations or highest order interactions in model
dim	numeric vector indicating the dimension of var (must be same length as var)
SubsetCoding	allows a (character) type or a matrix to be assigned to variables for each element of suffconfigs, see examples
MakeSubsets	boolean, indicates whether or not to use subsets of suffconfigs (used as option in MarginalMatrix)

Details

The design matrix for a model $\mu_{ij} = \alpha + \beta_i + \gamma_j$, where i and j each have three possible values, would be: `Designmatrix(c(1,2),list(c(1),c(2)),c(3,3))`. For readability, the use of characters is recommended for variable names, e.g., `Designmatrix(c("A","B"),list(c("A"),c("B")),c(3,3))`. The probability vector is assumed to be a vectorized form of the probabilities in a table, such that the last variable changes value fastest, then the before last variable, etc. For example, the cells of a 2×3 table are arranged in vector form as (11,12,13,21,22,23). To achieve this, the appropriate way to vectorize a data frame `dat` is using `c(t(ftable(dat)))`.

The optional argument `SubsetCoding` is useful for e.g. specifying various regression models, a linear by nominal model, grouping categories of a variable, or omitting a category. `SubsetCoding` has default value "Automatic", which is the same as the value "Nominal". Other options are "Linear", "Quadratic", "Cubic", "Quartic", "Quintic", "Identity". The command `ConstraintMatrix` is often more useful than `DesignMatrix` for specification of models for use in `SampleStatistics`, `ModelStatistics` or `MarginalModelFit`.

Value

matrix

Author(s)

W. P. Bergsma <w.p.bergsma@lse.ac.uk>

References

Bergsma, W. P. (1997). *Marginal models for categorical data*. Tilburg, The Netherlands: Tilburg University Press. http://stats.lse.ac.uk/bergsma/pdf/bergsma_phdthesis.pdf

Bergsma, W. P., Croon, M. A., & Hagenaars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. Berlin: Springer.

See Also

[ConstraintMatrix](#), [MarginalMatrix](#), [DirectSum](#)

Examples

```

# Design matrix for independence model
var <- c("A","B")
suffconfigs <- list(c("A"),c("B"))
dim <- c(3, 3)
DesignMatrix(var,suffconfigs,dim)
# notation in one line
DesignMatrix(c("A","B"),list(c("A"),c("B")),c(3,3))

# Design matrix for saturated model, two short specifications giving same result
DesignMatrix(c("A","B"),c("A","B"),c(3,3))
DesignMatrix(c("A","B"),list(c("A","B")),c(3,3))

# Design matrix for univariate quadratic regression model
var <- c("A")
suffconfigs <- c("A")
dim <- c(5)
DesignMatrix(var,suffconfigs,dim,SubsetCoding=list(c("A"),"Quadratic"))
# notation in one line
DesignMatrix(c("A"),c("A"),c(5),SubsetCoding=list(c("A"),"Quadratic"))

# Design matrix for linear by nominal model, various methods:
# simplest method which assumes equidistant centered scores:
DesignMatrix(
  var = c("A","B"),
  suffconfigs = c("A", "B"),
  dim = c(3,3),
  SubsetCoding = list(c("A","B"),list("Linear","Nominal")))
# alternative specification with same result as above:
DesignMatrix(
  var = c("A", "B"),
  suffconfigs = c("A", "B"),
  dim = c(3, 3),
  SubsetCoding = list(c("A","B"),list(rbind(c(-1,0,1)),rbind(c(1,0,0),c(0,1,0)))))
# specifying your own category scores
scores <- c(1,2,5);
DesignMatrix(
  var = c("A","B"),
  suffconfigs = c("A","B"),
  dim = c(3, 3),
  SubsetCoding = list(c("A","B"), list(rbind(scores), "Nominal")))

# Design matrix for nominal by nominal model, equating parameters
# of last two categories of second variable:
DesignMatrix(
  var = c("A", "B"),
  suffconfigs = c("A","B"),
  dim = c(3,3),

```

```
SubsetCoding = list(c("A", "B"), list("Nominal", rbind(c(1, 0, 0), c(0, 1, 1))))
```

DirectSum

DirectSum

Description

Returns the direct sum of two matrices.

Usage

```
DirectSum(...)
```

Arguments

... List of one or more matrices

Details

Standard mathematical direct sum operator.

Value

matrix

Author(s)

W. P. Bergsma <w.p.bergsma@lse.ac.uk>

References

Bergsma, W. P. (1997). *Marginal models for categorical data*. Tilburg, The Netherlands: Tilburg University Press. http://stats.lse.ac.uk/bergsma/pdf/bergsma_phdthesis.pdf

Bergsma, W. P., Croon, M. A., & Hagenars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. Berlin: Springer.

See Also

[ConstraintMatrix](#), [DesignMatrix](#), [DirectSum](#)

Examples

```
a <- matrix(1:12,3,4)
DirectSum(a) #returns a
DirectSum(a,a) #returns direct sum of a and a
DirectSum(a,a,a) #returns direct sum of a, a and a
```

DutchConcern

Concern about crime and social security in the Netherlands

Description

Data from a trend study where two survey questions, regarding (i) concern about crime and (ii) concern about social security, were asked to randomly selected people in the Netherlands at ten different time points (November 1977 to July 1981). The data are tabulated in Bergsma, Croon, and Hagenaaars (2009, Table 4.1, Table 4.2).

Section 4.1 in Bergsma, Croon, and Hagenaaars (2009).

Usage

```
data(DutchConcern)
```

Format

A data frame with 5742 observations on the following variables.

S Concern about social security (ordered): 1 = No (big) problem; 2 = big problem; 3 = Very big problem.

C Concern about crime (ordered): 1 = No (big) problem; 2 = big problem; 3 = Very big problem.

T time points (factor): 1 = Nov 1977; 2 = Jan 1978; 3 = Jun 1978; 4 = Nov 1978; 5 = Mar 1979; 6 = Oct 1979; 7 = Apr 1980; 8 = Oct 1980; 9 = Dec 1980; 10 = Jan 1981.

Source

Hagenaaars (1990, p. 289) and Continuous survey, University of Amsterdam / Steinmetz Archives Amsterdam.

References

Bergsma, W. P., Croon, M. A., & Hagenaaars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. Berlin: Springer

Hagenaaars, J. A. P. (1990). *Categorical longitudinal data: Log-linear panel, trend, and cohort analysis*. Newbury Park, CA: Sage.

Examples

```
data(DutchConcern)
n=c(t(ftable(DutchConcern)))

# Table 4.2
#NB: PLEASE REMOVE THE "#" BEFORE APPLY IN NEXT LINES, WON'T GO THROUGH R-CHECK OTHERWISE/
at1 = MarginalMatrix(c("S", "C", "T"), c("S", "T"), c(3, 3, 10));
print("Concern about social security:")
#apply(matrix(at1 %*% n, 10),1,function(x){100*x/sum(x)})
```

```

at2 = MarginalMatrix(c("S", "C", "T"), c("C", "T"), c(3, 3, 10));
print("Concern about crime:")
#apply(matrix(at2 %*% n, 10),1,function(x){100*x/sum(x)})

#####
# Define and fit models for marginal table IRT (Section 4.1.1)

# atIRT %*% n produces IRT table, dimension 2x3x10
atIRT = MarginalMatrix(c("S", "C", "T"), list(c("S", "T"), c("C", "T")), c(3, 3, 10));
# bt1.Log(atIRT.pi)=0 is marginal model for independence of IT and R \
bt1 = ConstraintMatrix(c("I", "R", "T"), list(c("I", "T"), c("R")), c(2, 3, 10));
bt2 = ConstraintMatrix(c("I", "R", "T"), list(c("I", "T"), c("R", "T")), c(2, 3, 10));
bt3 = ConstraintMatrix(c("I", "R", "T"), list(c("I", "T"), c("I", "R")), c(2, 3, 10));
bt4 = ConstraintMatrix(c("I", "R", "T"), list(c("I", "T"),
      c("I", "R"), c("R", "T")), c(2, 3, 10));

model1 = list(bt1, "log", atIRT);
model2 = list(bt2, "log", atIRT);
model3 = list(bt3, "log", atIRT);
model4 = list(bt4, "log", atIRT);

# change model1 to model2 etc to fit different models
pi1 = MarginalModelFit(n, model4,
  ShowProgress = 5,
  CoefficientDimensions = c(2, 3, 10),
  Labels = c("I", "R", "T"));

#####
# Simultaneous model for marginal and joint distributions (Section 4.1.2)

# define x, the design matrix for the loglinear model of Eq. 4.1
x <- DesignMatrix(var = c("S", "C", "T"),
  suffconfigs = c("S", "C", "T"),
  dim = c(3,3,10),
  SubsetCoding = list(c("S", "C", "T"),list("Nominal","Nominal","Linear")))
# model6 is the simultaneous model
model6 <- list(model4, x);

# NB: when fitting model6 Labels and CoefficientDimensions should be appropriate
# to get the right output # for table IRT, which is different than for model5
#NB: REMOVE "#" IN NEXT LINE, WON'T GO THROUGH R-CHECK
#pi5 = MarginalModelFit(DutchConcern, model6, ShowProgress = 5,
# CoefficientDimensions = c(2, 3, 10), Labels = c("I", "R", "T"), MaxSteps = 500, MaxStepSize=.2)

```


Description

The data come from a Dutch panel study (T1 = February 1977, T2 = March 1977) and concern the questions for which party the respondent intends to vote (variables A and B, respectively) and which candidate the respondent prefers to become the next Prime Minister (C and D). The data have been analyzed before (Hagenaars, 1986, 1988, 1990), and more information on the panel study and the outcomes may be obtained from these references.

The data are tabulated in Bergsma, Croon, and Hagenaars (2009, Table 6.1).

Usage

```
data(DutchPolitics)
```

Format

A data frame with 1100 observations on the following variables.

A Party preference at time 1 (factor): 1 = Christian Democrats; 2 = Left wing; 3 = Other.

B Party preference at time 2 (factor): 1 = Christian Democrats; 2 = Left wing; 3 = Other.

C Candidate preference at time 1 (factor): 1 = Christian Democrats; 2 = Left wing; 3 = Other.

D Candidate preference at time 2 (factor): 1 = Christian Democrats; 2 = Left wing; 3 = Other.

Source

J. A. Hagenaars (1990). *Categorical longitudinal data: log-linear, panel, trend, and cohort analysis*. Newbury Park: Sage

References

Bergsma, W. P., Croon, M. A., & Hagenaars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. Berlin: Springer

J. A. Hagenaars (1990). *Categorical longitudinal data: log-linear, panel, trend, and cohort analysis*. Newbury Park: Sage

Examples

```
data(DutchPolitics)

# Marginal homogeneity: A=C and B=D
at2a <- MarginalMatrix(c("A", "B", "C", "D"), list(c("A"), c("C")), c(3, 3, 3, 3));
at2b <- MarginalMatrix(c("A", "B", "C", "D"), list(c("B"), c("D")), c(3, 3, 3, 3));
bt2 <- ConstraintMatrix(c(1,2), list(c(1),c(2)), c(2,3));
at2 <- rbind(at2a, at2b);
bt2 <- DirectSum(bt2, bt2);

model <- list(bt2, "identity", at2);

mpolMH <- MarginalModelFit(DutchPolitics, model,
  MaxError = 10.^-25,
  MaxSteps = 200,
```

```
MaxStepSize = .5,  
StartingPoint = "Automatic",  
CoefficientDimensions = c(2, 2, 3),  
ShowProgress = 50);
```

ErieCounty

Erie County political preference, two-wave panel

Description

These data come from the first systematic panel study on voting, conducted by Lazarsfeld and his associates in Erie County, Ohio in 1940 (Lazarsfeld et al, 1948; Lazarsfeld, 1972, Wiggins, 1973, Hagenaars, 1993). The data are presented in Table 6.3 and refer to the variables *A* – *Party preference at time 1* – August 1940 (1.\ Republican 2.\ Democrat), *B* – *Presidential Candidate preference at time 1* (1.\ for Willkie 2.\ against Willkie), *C* – *Party preference at time 2* – October 1940, and *D* – *Presidential Candidate preference at time 2*. Wendell Willkie was the (defeated) 1940 Republican presidential candidate running against the Democrat Franklin D. Roosevelt.

Section 6.3 in Bergsma, Croon, and Hagenaars (2009)

Usage

```
data(ErieCounty)
```

Format

A data frame with 266 observations on the following variables.

A Party Preference T_1 (August 1940): 1 = Democrat; 2 = Republican;

B Candidate Preference T_1 (August 1940): 1 = for Willkie; 2 = against Willkie;

C Party Preference T_2 (October 1940): 1 = Democrat; 2 = Republican;

D Candidate Preference T_2 (October 1940): 1 = for Willkie; 2 = against Willkie;

Source

CBS News and New York Times 2001.

References

Bergsma, W. P., Croon, M. A., & Hagenaars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. Berlin: Springer

Examples

```
data(ErieCounty)
```

EVS *European Values Study (EVS): attitude towards women's role in society*

Description

European Values Study 1999/2000, Views on Women's Roles <http://www.europeanvaluesstudy.eu/>

The data are tabulated in Bergsma, Croon, and Hagenaars (2009, Table 5.1a). Section 5.1.2 in Bergsma, Croon and Hagenaars (2009).

Usage

```
data(EVS)
```

Format

A data frame with 960 observations on the following variables.

S Sex (factor): 1 = Male; 2 = Female.

A Date of Birth (ordered): 1 = Before 1945; 2 = 1945-1963; 3 = After 1963.

E Level of education (ordered): 1 = Lower; 2 = Intermediate; 3 = Higher.

R Religion (ordered): 1 = Religious person; 2 = Not a religious person; 3 = Convinced atheist.

W Attitude women's role in society (factor): 1 = Traditional; 2 = Nontraditional.

Source

European Values Study 1999/2000

References

Bergsma, W. P., Croon, M. A., & Hagenaars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. New York: Springer.

Examples

```
data(EVS)

# Table SAERW
var = c("S", "A", "E", "R", "W");
dim = c(2, 3, 3, 3, 2);

# matrices for table SA
at1 <- MarginalMatrix(var, c("S", "A"), dim);
bt1 <- ConstraintMatrix(c("S", "A"), list(c("S"), c("A")), c(2, 3));

# matrices for table SAER
```

```

at2 <- MarginalMatrix(var, c("S", "A", "E", "R"), dim);
bt2 <- ConstraintMatrix(var = c("S", "A", "E", "R"),
  suffconfigs = list(c("S", "A", "E"), c("S", "R"), c("A", "R")),
  dim = c(2, 3, 3, 3));

# matrices for table SAERW: constraints
at3 <- MarginalMatrix(var, c("S", "A", "E", "R", "W"), dim);
bt3 <- ConstraintMatrix(var = c("S", "A", "E", "R", "W"),
  suffconfigs = list(c("S", "A", "E", "R"), c("S", "W"), c("A", "W"), c("E", "W"), c("R", "W")),
  dim = c(2, 3, 3, 3, 2))

# matrix for table SAERW: design matrix
x <- DesignMatrix(var = c("S", "A", "E", "R", "W"),
  suffconfigs = list(c("S", "A", "E", "R"), c("S", "W"), c("A", "W"), c("E", "W"), c("R", "W")),
  dim = c(2, 3, 3, 3, 2));

# model1: specification using only constraints
at <- rbind(at1, at2, at3);
bt <- DirectSum(bt1, bt2);
bt <- DirectSum(bt, bt3);
model1 <- list(bt, "log", at);

# model2: same as model1 but using both constraints and a design matrix
# to specify a loglinear model for the joint distribution
at <- rbind(at1, at2);
bt <- DirectSum(bt1, bt2);
model2 <- list(list(bt, "log", at), x);

nkps3 <- MarginalModelFit(EVS, model2, MaxError = 10.^-25,
  MaxSteps = 1000,
  ShowProgress = 10,
  MaxStepSize = .3 );

```

GSS93

Political Orientation and Religion in the United States in 1993 (General Social Survey, 1993)

Description

Self-reported Political Orientation (P), Religion (R), and Opinion of Teenage Birth-control (B) of a sample of 911 US citizens in 1993. The data stem from the General Social Survey. The data are tabulated in Bergsma, Croon, and Hagenaars (2009, Table 2.1, Table 2.3).

See Section~2.1 in Bergsma, Croon, and Hagenaars (2009).

Several worked examples involving this data set are listed below but more can be found at <http://stats.lse.ac.uk/bergsma/cmm/Rfiles/GSS93.R>

Usage

```
data(GSS93)
```

Format

A data frame with 911 observations on the following three variables.

P Political orientation (ordered): 1 = Extremely liberal; 2 = Liberal; 3 = Slightly liberal; 4 = Moderate; 5 = Slightly conservative; 6 = Conservative; 6 = Extremely conservative.

R Religion (factor): 1 = Protestant; 2 = Catholic; 3 = Other.

B Opinion of birth control (ordered): 1 = Strongly agree; 2 = Agree; 3 = Disagree; 4 = Strongly disagree;

Source

General Social Survey (1993)

References

Bergsma, W. P., Croon, M. A., & Hagnaars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. New York: Springer

General Social Survey (1993).

Examples

```
data(GSS93)

## Table 2.1 of Bergsma, Croon, & Hagnaars (2009)
addmargins(table(GSS93[,1:2]))

## Table 2.2 of Bergsma, Croon, & Hagnaars (2009)
# Specify coefficients
coeff <- list("log",diag(21))
SampleStatistics(dat = GSS93[, 1 : 2],
  coeff = coeff,
  CoefficientDimensions = c(7, 3),
  Labels = c("P", "R"),
  ShowParameters = TRUE,
  ShowCoefficients = FALSE)

## Table 2.3 of Bergsma, Croon, & Hagnaars (2009)
ftable(B + R ~ P , data = GSS93)

#####
## Models for table PR

#independence of P and R
bt1 <- ConstraintMatrix(c("P", "R"), list(c("P"), c("R")), c(7,3));
#linear by nominal model
bt2 <- ConstraintMatrix(var = c("P", "R"),
  suffconfigs = list(c("P", "R")),
  dim = c(7, 3),
  SubsetCoding = list(c("P", "R"), c("Linear", "Nominal")))
```

```

#linear by nominal model with equality of first two nominal parameters
bt3 <- ConstraintMatrix(var = c("P", "R"),
  suffconfigs = list(c("P", "R")),
  dim = c(7, 3),
  SubsetCoding = list(c("P", "R"), list("Linear", rbind(c(1, 1, 0), c(0, 0, 1)))))
m <- MarginalModelFit(dat = GSS93[,1:2],
  model = list(bt2,"log"),
  ShowCoefficients = FALSE,
  ShowProgress = 1,
  ShowParameters = TRUE,
  CoefficientDimensions = c(7, 3),
  Labels = c("P", "R"),
  ParameterCoding = list("Polynomial", "Effect"))

#####
## Models for table PRB

#various loglinear models
bt1 <- ConstraintMatrix(c("P", "R","B"), list(c("P","R"),c("B")), c(7,3,4))
bt2 <- ConstraintMatrix(c("P", "R","B"), list(c("P","R"),c("R","B")), c(7,3,4))
bt3 <- ConstraintMatrix(c("P", "R","B"), list(c("P","R"),c("P","B")), c(7,3,4))
bt4 <- ConstraintMatrix(c("P", "R","B"), list(c("P","R"),c("P","B"),c("R","B")), c(7,3,4))
bt5 <- ConstraintMatrix(c("P", "R","B"), list(c("P","R"),c("P","B"),c("R","B")), c(7,3,4),
  SubsetCoding = list(list(c("P", "B"), c("Linear", "Linear")),
    list(c("R", "B"), c("Nominal", "Linear"))))

m <- MarginalModelFit(dat = GSS93,
  model = list(bt2,"log"),
  ShowCoefficients = FALSE,
  ShowProgress = 1,
  ShowParameters = TRUE,
  CoefficientDimensions =c(7, 3, 4),
  Labels = c("P", "R", "B"),
  ParameterCoding = list("Polynomial", "Polynomial", "Effect"))

```

JoinModels

JoinModels

Description

Returns the simultaneous specification of two models

Usage

```
JoinModels(...)
```

Arguments

... list of 'compatible' models, i.e., each model is specified using the same number of functions (and matrices)

Details

Models can be of any form allowed in CMM (see `MarginalModelFit`), eg, `list(bt,coeff,at)`, with the restriction that the number of columns of the `at` matrices must be equal, and the list of functions in `coeff` must be identical.

Value

CMM model form

Author(s)

W. P. Bergsma <w.p.bergsma@lse.ac.uk>

References

Bergsma, W. P. (1997). *Marginal models for categorical data*. Tilburg, The Netherlands: Tilburg University Press. http://stats.lse.ac.uk/bergsma/pdf/bergsma_phdthesis.pdf

Bergsma, W. P., Croon, M. A., & Hagenaars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. Berlin: Springer.

See Also

[DirectSum](#), [SpecifyCoefficient](#), [MarginalModelFit](#)

Examples

```
# simultaneously specify marginal independence in two marginal tables

bt1 = ConstraintMatrix(c("A", "B"), list(c("A"), c("B")), c(3, 3))
at1 = MarginalMatrix(c("A", "B", "C"), c("A", "B"), c(3, 3, 3))
model1 = list(bt1, "log", at1)

bt2 = ConstraintMatrix(c("B", "C"), list(c("B"), c("C")), c(3, 3))
at2 = MarginalMatrix(c("A", "B", "C"), c("B", "C"), c(3, 3, 3))
model2 = list(bt2, "log", at2)

model12 = JoinModels(model1, model2)

# the model can be fitted to an artificial data set
n = c(1:27)
fit = MarginalModelFit(n, model12)
```

LaborParticipation *Women's labor participation: 1967-1971*

Description

The labor participation (yes/no) of 1583 women was measured in five consecutive year, 1967-1971, leading to a $2 \times 2 \times 2 \times 2 \times 2$ table.

The data are tabulated in Bergsma, Croon, and Hagenaars (2009, p. 128).

Section 4.3 in Bergsma, Croon and Hagenaars, 2009

Usage

```
data(LaborParticipation)
```

Format

A data frame with 1583 observations on the following variables.

Year1967 Participated in 1967 (factor): 1 = No; 2 = Yes.

Year1968 Participated in 1968 (factor): 1 = No; 2 = Yes.

Year1969 Participated in 1969 (factor): 1 = No; 2 = Yes.

Year1970 Participated in 1970 (factor): 1 = No; 2 = Yes.

Year1971 Participated in 1971 (factor): 1 = No; 2 = Yes.

Source

Heckman & Willis (1977).

References

Bergsma, W. P., Croon, M. A., & Hagenaars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. New York: Springer.

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(LaborParticipation)
n <- c(t(ftable(LaborParticipation)))

#####
#Sample kappa values

#matrix for obtaining transition matrices for consecutive years
at <- MarginalMatrix(var = c("67", "68", "69", "70", "71"),
  marg = list(c("67", "68"), c("68", "69"), c("69", "70"), c("70", "71"))),
```



```

dim = c(2, 2, 2, 2, 2))

coeff <- SpecifyCoefficient("CohenKappa", arg = 2, rep = 4);
stats <- SampleStatistics(n, list(coeff,at), ShowParameters = FALSE)

#####
#Fitting models for kappa
#matrix for obtaining transition matrices for consecutive years
at <- MarginalMatrix(var = c("67", "68", "69", "70", "71"),
  marg = list(c("67", "68"), c("68", "69"), c("69", "70"), c("70", "71")),
  dim = c(2, 2, 2, 2, 2))

coeff <- SpecifyCoefficient("CohenKappa", arg = 2, rep = 4);

bt1 <- ConstraintMatrix(var = c(1), suffconfigs = c(), dim = c(4)); #equal kappas
bt2 <- rbind(c(1,-2,1,0), c(0,1,-2,1)); #linear trend for kappas
model <- list(bt1, coeff,at)

m = MarginalModelFit(n, model, ShowParameters = FALSE, ShowProgress = 10)

```

MarginalMatrix

MarginalMatrix

Description

Returns marginal matrix; i.e., matrix required to obtained marginal frequencies

Usage

```
MarginalMatrix(var, marg, dim, SubsetCoding="Identity",SelectCells="All")
```

Arguments

var	character or numeric vector containing variables
marg	list of character or numeric indicating marginals
dim	numeric vector indicating the dimension of var
SubsetCoding	allows a (character) type or a matrix to be assigned to variables for each element of suffconfigs, see examples and DesignMatrix
SelectCells	Useful option for empirical likelihood. Default "All" includes all columns, if a list of cells in the table is given only the corresponding columns of Marginal-Matrix are included.

Details

Gives the matrix which, multiplied by a probability vector, gives the marginal probabilities. The probability vector is assumed to be a vectorized form of the probabilities in a table, such that the last variable changes value fastest, then the before last variable, etc. For example, the cells of a 2×3 table are arranged in vector form as (11,12,13,21,22,23). To achieve this, the appropriate way to vectorize a data frame `dat` is using `c(t(ftable(dat)))`.

Special case of transposed `DesignMatrix`:

```
MarginalMatrix <- function(var,marg,dim,SubsetCoding="Identity")
  t(DesignMatrix(var,marg,dim,SubsetCoding=SubsetCoding,MakeSubsets=FALSE))
```

Allows weighted sums of probabilities using `SubsetCoding`

Value

matrix

Author(s)

W. P. Bergsma <w.p.bergsma@lse.ac.uk>

References

Bergsma, W. P. (1997). *Marginal models for categorical data*. Tilburg, The Netherlands: Tilburg University Press. http://stats.lse.ac.uk/bergsma/pdf/bergsma_phdthesis.pdf

Bergsma, W. P., Croon, M. A., & Hagenaars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. Berlin: Springer.

See Also

[ConstraintMatrix](#), [DesignMatrix](#), [DirectSum](#)

Examples

```
# Computing marginal frequencies
n <- c(1:6) #example list of frequencies
var <- c("A","B")
marg <- list(c("A"),c("B"))
dim <- c(2,3)
at <- MarginalMatrix(var,marg,dim)
# list of marginal frequencies:
at

# identitymatrix: several ways of specifying:
marg <- c("A","B")
MarginalMatrix(var, marg,dim)
MarginalMatrix(var, marg, dim,
  SubsetCoding = list(c("A", "B"), list("Identity", "Identity")))
MarginalMatrix(var, marg, dim,
  SubsetCoding = list(c("A", "B"), list(rbind(c(1,0),c(0,1)), rbind(c(1,0,0),c(0,1,0),c(0,0,1)))))
```

```
# omit second category of first variable
at <- MarginalMatrix(var, marg, dim,
  SubsetCoding = list(c("A","B"), list(rbind(c(1,0)), "Identity")))
at
```

MarginalModelFit

MarginalModelFit

Description

Fits marginal models, by default using maximum likelihood.

Usage

```
MarginalModelFit(dat, model, ShowSummary = TRUE, MaxSteps = 1000, MaxStepSize = 1,
  MaxError = 1e-20, StartingPoint = "Automatic", MaxInnerSteps = 2,
  ShowProgress = TRUE, CoefficientDimensions="Automatic", Labels="Automatic",
  ShowCoefficients = TRUE, ShowParameters = FALSE, ParameterCoding = "Effect",
  ShowCorrelations = FALSE, Method = "ML", Title = "Summary of model fit")
```

Arguments

<code>dat</code>	vector of frequencies or data frame
<code>model</code>	list specified eg as <code>list(bt, coeff, at)</code>
<code>ShowSummary</code>	Whether or not to execute <code>summary()</code> of the output
<code>MaxSteps</code>	integer: maximum number of steps of the algorithm
<code>MaxStepSize</code>	number greater than 0 and at most 1: step size
<code>MaxError</code>	numeric: maximum error term
<code>StartingPoint</code>	vector of starting frequencies corresponding to all cells in the manifest table
<code>MaxInnerSteps</code>	nonnegative integer: only used for latent variable models, indicates number of steps in M step of EM algorithms
<code>ShowProgress</code>	boolean or integer: FALSE for no progress information, TRUE or 1 for information at every step, an integer k for information at every k-th step
<code>CoefficientDimensions</code>	numeric vector of dimensions of the table in which the coefficient vector is to be arranged
<code>Labels</code>	list of characters or numbers indicating labels for dimensions of table in which the coefficient vector is to be arranged
<code>ShowCoefficients</code>	boolean, indicating whether or not the coefficients are to be displayed
<code>ShowParameters</code>	boolean, indicating whether or not the parameters (computed from the coefficients) are to be displayed

ParameterCoding	Coding to be used for parameters, choice of "Effect", "Dummy" and "Polynomial"
ShowCorrelations	boolean, indicating whether or not to show the correlation matrix for the estimated coefficients
Method	character, choice of "ML" for maximum likelihood or "GSK" for the GSK method
Title	title of computation to appear at top of screen output

Details

The data can be a data frame or vector of frequencies. `MarginalModelFit` converts a data frame `dat` to a vector of frequencies using `c(t(ftable(dat)))`.

The model specification is fairly flexible. We first describe the most typical way to specify the model. The model itself should typically first be written in the form of a constraint vector as $B'\theta(A'\pi) = 0$ where B' is a contrast matrix, A' is matrix, normally of zeroes and ones, such that $A'pi$ gives a vector of marginal probabilities, and the function `theta` yields a list of (marginal) *coefficients*. The model is then specified as `model=list(bt,coeff,at)` where `bt` is the matrix B' , `at` is the matrix A' , and `coeff` represents the vector of coefficients using the generalized exp-log notation. For most of the models in the book, `bt` can be obtained directly using `ConstraintMatrix`, `coeff` can be obtained directly using `SpecifyCoefficient`, and `at` can be obtained directly using `MarginalMatrix`.

Note that CMM does not permit the C and X matrix in the model $C'\theta(A'\pi) = X\beta$ to be specified for use in the programme. The model has to be rewritten in terms of constraints as above, which is normally straightforward to do with the use of `ConstraintMatrix`. For many purposes, estimates and standard errors for a beta vector as in the equation above can still be obtained using the optional argument `ShowParameters=TRUE`.

There are two ways to specify `coeff`. The first is using the generalized exp-log notation, in which case `coeff[[1]]` should be a list of matrices, and `coeff[[2]]` should be a list of predefined functions of the same length. The second is to set `coeff` equal to a predefined function; for example, marginal loglinear models are obtained by setting `coeff="log"`.

The model can be specified in various other ways: as `model=list(coeff,at)`, `model=list(bt,coeff)`, `model=at`, or even just `model=coeff`. Furthermore, the model $B'\theta(A'\pi) = d$ with d a nonzero vector is specified in the form `model=list(bt,coeff,at,d)`.

To specify the simultaneous model $B'\theta(A'\pi) = 0$

$\log \pi = X\beta$ the extended model specification `model=list(margmodel,x)` should be used, where `margmodel` has one of the above forms, and `x` is a design matrix, which can be obtained using `DesignMatrix`. Fitting is often more efficient by specifying a loglinear model for the joint distribution in this way rather than using constraints.

The default on-screen output when running `fit=MarginalModelFit(...)` is given by `summary(fit)`. Important here is the distinction between coefficients and parameters, briefly described above. Standard output gives the coefficients. These are that part of `model` without the `bt` matrix, eg if the model is `list(bt,coeff,at)` then the coefficients are `list(coeff,at)`. If other coefficients are needed, [ModelStatistics](#) can be used.

Latent variable models can be specified: if the size of the table for which `model` is specified is a multiple of the the size of the observed frequencies specified in `dat`, it is assumed this is due to

the presence of latent variables. With respect to vectorization, the latent variables are assumed to change their value fastest.

Convergence may not always be achieved with `MaxStepSize=1` and a lower value may need to be used, but not too low or convergence is slow. If the step size is too large, a typical error message is "system is computationally singular: reciprocal condition number = 1.35775e-19"

Value

Most of the following are included in any output. Use `summary()` to get a summary of output.

`FittedFrequencies`

Vector of fitted frequencies for the full table (including any latent variables).

`Method`

Fitting method used (currently maximum likelihood, GSK or minimum discrimination information)

`LoglikelihoodRatio`

`ChiSquare`

`DiscriminationInformation`

`WaldStatistic`

`DegreesOfFreedom`

`PValue`

p-value based on asymptotic chi-square approximation for likelihood ratio test statistic

`SampleSize`

`BIC`

`Eigenvalues`

`ManifestFittedFrequencies`

For the "coefficients" in the equation $bt.coef(at.pi)=d$, the following statistics are available:

`ObservedCoefficients`

`FittedCoefficients`

`CoefficientStandardErrors`

`CoefficientZScores`

`CoefficientAdjustedResiduals`

`CoefficientCovarianceMatrix`

`CoefficientCorrelationMatrix`

CoefficientAdjustedResidualCovarianceMatrix

CoefficientDimensions

CoefficientTableVariableLabels

CoefficientTableCategoryLabels

The “parameters” are certain linear combinations of the coefficients. For example, if the coefficients are log probabilities, then the parameters are the usual loglinear parameters.

Parameters

For the i -th subset of variables, the parameters are obtained by

Parameters[[i]]\$

The following statistics for the parameters belonging to each subset of variable are available.

Parameters[[i]]\$ObservedCoefficients

Parameters[[i]]\$FittedCoefficients

Parameters[[i]]\$CoefficientStandardErrors

Parameters[[i]]\$CoefficientZScores

Parameters[[i]]\$CoefficientAdjustedResiduals

Parameters[[i]]\$CoefficientCovarianceMatrix

Parameters[[i]]\$CoefficientCorrelationMatrix

Parameters[[i]]\$CoefficientAdjustedResidualCovarianceMatrix

Parameters[[i]]\$CoefficientDimensions

Parameters[[i]]\$CoefficientTableVariableLabels

Parameters[[i]]\$CoefficientTableCategoryLabels

Author(s)

W. P. Bergsma <w.p.bergsma@lse.ac.uk>

References

Bergsma, W. P. (1997). *Marginal models for categorical data*. Tilburg, The Netherlands: Tilburg University Press. http://stats.lse.ac.uk/bergsma/pdf/bergsma_phdthesis.pdf

See Also

[SampleStatistics](#), [ModelStatistics](#)

Examples

```
# see also the built-in data sets

data(NKPS)

# Fit the model asserting Goodman and Kruskal's gamma is zero for
# Child's attitude toward sex role's (NKPS[,3], three categories) and
# parent's attitude toward sex role's (NKPS[,4], three categories).

coeff = SpecifyCoefficient("GoodmanKruskalGamma",c(3,3))
fit = MarginalModelFit(NKPS[,c(3,4)], coeff )

# Marginal homogeneity (MH) in a 3x3 table AB
# Note that MH is equivalent to independence in the 2x3 table of marginals IR, in which
# the row with I=1 gives the A marginal, and the row with I=2 gives the B marginal
n <- c(1,2,3,4,5,6,7,8,9)
at <- MarginalMatrix(c("A","B"),list(c("A"),c("B")),c(3,3))
bt <- ConstraintMatrix(c("I","R"),list(c("I"),c("R")),c(2,3))
model <- list( bt, "log", at)
fit <- MarginalModelFit(n,model)

#Output can be tidied up:
fit <- MarginalModelFit(n,model,CoefficientDimensions=c(2,3))
```

MarihuanaAlcohol

Marihuana and alcohol use during adolescence, five-wave panel

Description

Panel study with five time points. A group of 269 youths were interviewed at ages 13, 14, 15, 16, and 17, and asked (among other things) about their marijuana and alcohol use (Eliot, Huizinga & Menard, 1989). The data are tabulated in Bergsma, Croon, and Hagenaars (2009, p. 128). 208 observations do not have missing values.

Sections 4.2 and 4.4 in Bergsma, Croon, and Hagenaars (2009).

Usage

```
data(MarihuanaAlcohol)
```

Format

A data frame with 269 observations on the following variables.

Gender (factor): 1 = Male; 2 = Female.

M1 Use of marihuana at time 1 (ordered): 1 = Never; 2 = Occasionally; 3 = Frequently.

M2 Use of marihuana at time 2 (ordered): see M1.

M3 Use of marihuana at time 3 (ordered): see M1.

M4 Use of marihuana at time 4 (ordered): see M1.

M5 Use of marihuana at time 5 (ordered): see M1.

A1 Use of alcohol at time 1 (ordered): see M1.

A2 Use of alcohol at time 2 (ordered): see M1.

A3 Use of alcohol at time 3 (ordered): see M1.

A4 Use of alcohol at time 4 (ordered): see M1.

A5 Use of alcohol at time 5 (ordered): see M1.

Source

US National Youth Survey (NYS): teenage marijuana and alcohol use (Elliot, Huizinga and Menard, 1989)

References

Bergsma, W. P., Croon, M. A., & Hagenaars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. New York: Springer.

Elliot, D. S., Huizinga, D. & Menard, S. (1989). *Multiple problem youth: Delinquency, substance use, and mental health problems*. New York: Springer.

Examples

```
data(MarihuanaAlcohol)

# Table MA: marginal loglinear analysis (BCH Section 4.2.1)
# listwise deletion of missing values and deletion of Gender and Alcohol use
dat <- MarihuanaAlcohol[-row(MarihuanaAlcohol)[is.na(MarihuanaAlcohol)],2:6]

# at yields the vectorized 5x3 table MA (marijuana use x age)
at <- MarginalMatrix(var = c("M1", "M2", "M3", "M4", "M5"),
  marg = list(c("M1"), c("M2"), c("M3"), c("M4"), c("M5")),
  dim = c(3, 3, 3, 3, 3))
obscoeff <- SampleStatistics(dat = dat,
  coeff = list("log", at),
  CoefficientDimensions = c(5,3),
  Labels = c("Age", "M"),
  ShowCoefficients = FALSE,
  ShowParameters = TRUE)
```

ModelStatistics	<i>ModelStatistics</i>
-----------------	------------------------

Description

If fitted frequencies under a model have been calculated, this procedure can be used to give sample values, fitted values, estimated standard errors, z-scores and adjusted residuals of one or more coefficients.

Usage

```
ModelStatistics(dat, fitfreq, model, coeff, CoefficientDimensions = "Automatic",
  Labels = "Automatic", Method = "ML", ShowCoefficients = TRUE, ShowParameters = FALSE,
  ParameterCoding = "Effect", ShowCorrelations = FALSE, Title = "")
```

Arguments

dat	observed data as a list of frequencies or as a data frame
fitfreq	vector of fitted frequencies for each cell of full table (including latent variables, if any)
model	list specified eg as <code>list(bt, coeff, at)</code>
coeff	list of coefficients, can be obtained using SpecifyCoefficient
CoefficientDimensions	numeric vector of dimensions of the table in which the coefficient vector is to be arranged
Labels	list of characters or numbers indicating labels for dimensions of table in which the coefficient vector is to be arranged
ShowCoefficients	boolean, indicating whether or not the coefficients are to be displayed
ShowParameters	boolean, indicating whether or not the parameters (computed from the coefficients) are to be displayed
Method	character, choice of "ML" for maximum likelihood or "GSK" for the GSK method
ParameterCoding	Coding to be used for parameters, choice of "Effect", "Dummy" and "Polynomial"
ShowCorrelations	boolean, indicating whether or not to show the correlation matrix for the estimated coefficients
Title	title of computation to appear at top of screen output

Details

The data can be a data frame or vector of frequencies. `MarginalModelFit` converts a data frame `dat` using `c(t(ftable(dat)))`.

For `ParameterCoding`, the default for "Dummy" is that the first cell in the table is the reference cell. Cell (i, j, k, \dots) can be made reference cell using `list("Dummy", c(i, j, k, \dots))`. For "Polynomial" the default is to use centralized scores based on equidistant (distance 1) linear scores, for example, if for $i = 1, 2, 3, 4$, $\mu_i = \alpha + q_i\beta + r_i\gamma + s_i\delta$ where β is a quadratic, γ a cubic and δ a quartic effect, then q_i takes the values $(-1.5, -.5, .5, 1.5)$, r_i takes the values $(1, -1, -1, 1)$ (centralized squares of the q_i), and s_i takes the values $(-3.375, -.125, .125, 3.375)$ (cubes of the q_i).

Value

A subset of the output of `MarginalModelFit`.

Author(s)

W. P. Bergsma <w.p.bergsma@lse.ac.uk>

References

Bergsma, W. P. (1997). *Marginal models for categorical data*. Tilburg, The Netherlands: Tilburg University Press. http://stats.lse.ac.uk/bergsma/pdf/bergsma_phdthesis.pdf

Bergsma, W. P., Croon, M. A., & Hagenars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. Berlin: Springer.

See Also

[ModelStatistics](#), [MarginalModelFit](#)

Examples

```
# Below an example where ModelStatistics can be used to get output for coefficients
# not given by MarginalModelFit

# Marginal homogeneity (MH) in a 3x3 table AB
# Note that MH is equivalent to independence in the 2x3 table of marginals IR, in which the
# row with I=1 gives the A marginal, and the row with I=2 gives the B marginal
n <- 1 : 9
at <- MarginalMatrix(c("A", "B"), list(c("A"), c("B")), c(3,3))
bt <- ConstraintMatrix(c("I", "R"), list(c("I"), c("R")), c(2,3))
model <- list( bt, "log", at)

#The "coefficients" for the model are the log marginal probabilities for table IR
fit <- MarginalModelFit(dat = n, model = model,
  CoefficientDimensions = c(2, 3), Labels = c("I", "R"))

#to get output for the marginal probabilities, ModelStatistics can be used as follows
spec <- SpecifyCoefficient("ConditionalProbabilities",list(c("I", "R"),c("I"),c(2,3)))
coeff <- list(spec, at)
```

```
stats <- ModelStatistics(dat = n, fitfreq = fit$FittedFrequencies,
  model = model, coeff = coeff, CoefficientDimensions = c(2, 3),
  Labels = c("I", "R"))
```

NES

*Political Orientation in the US, three-wave panel study***Description**

Data from the US National Election Studies. Three-wave panel study measuring political orientation on a seven-point scale. The data are tabulated in Bergsma, Croon, and Hagenaaars (2009, 4.4).

Sections 4.2.1 and 4.3 in Bergsma, Croon and Hagenaaars (2009).

Usage

```
data(NES)
```

Format

A data frame with 408 observations on the following variables.

T1 Political orientation at time 1 (ordered): 1 = Extremely liberal 2 = Liberal 3 = Slightly liberal" 4 = Moderate 5 = Slightly conservative 6 = Conservative 7 = Extremely conservative

T2 Political orientation at time 2 (ordered): see T1

T3 Political orientation at time 3 (ordered): see T1

Source

US National Election Studies.

References

Bergsma, W. P., Croon, M. A., & Hagenaaars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. New York: Springer.

Examples in book: <http://stats.lse.ac.uk/bergsma/cmm/Rfiles/NES.R>

Examples

```
data(NES)
```

```
#####
# Models for marginal homogeneity over time (Section 4.2.1)
```

```
# Marginal homogeneity : no change in political orientation over time
```

```
at <- MarginalMatrix(c("T1", "T2", "T3"), list(c("T1"), c("T2"), c("T3")), c(7,7,7));
bt1 <- ConstraintMatrix(c("T", "P"), list(c("T"), c("P")), c(3, 7));
```

```

model1 <- list(bt1, "identity", at);

start <- c(t(ftable(NES))) + .001;
pihat <- MarginalModelFit(NES, model1,
  MaxSteps = 1000, StartingPoint = start,
  ShowProgress = 100, MaxError = 1e-28,
  CoefficientDimensions = c(3,7), ShowCoefficients = TRUE,
  ShowParameters = FALSE, Labels = c("T", "P"));

```

NKPS	<i>Attitudes on sex roles and marriage, measurements clustered in families</i>
------	--

Description

Netherlands Kinship Panel Study (NKPS), www.nkps.nl. Netherlands Kinship Panel Study (NKPS), a unique in-depth large-scale study into (changing) kinship relationships covering a large number of life domains (Dykstra et al., 2004).

The data are tabulated in Bergsma, Croon, and Hageaars (2009, Table 2.8, 2.10, 2.11, 2.12). In Sections 5.1 and 6.4.2 more variables of the same data set are used, and these have only 1797 observations with no missing values; this set is available as NKPS2.

Sections 2.2.3, 3.2, 5.1, 6.4.2 in Bergsma, Croon and Hageaars (2009)

Usage

```

data(NKPS)
data(NKPS2)

```

Format

A data frame with 1884 observations on the following variables.

C Child's Gender (factor): 1 = Son 2 = Daughter

P Parent's Gender (factor): 1 = Father 2 = Mother

CS Child's sex role attitude (ordered): 1 = Nontrad. 2 = Mod. trad. 3 = Trad.

PS Parent's sex role attitude (ordered): 1 = Nontrad. 2 = Mod. trad. 3 = Trad.

CM Child's marriage attitude (ordered): 1 = Nontrad. 2 = Mod. trad. 3 = Trad.

PM Parent's marriage attitude (ordered): 1 = Nontrad. 2 = Mod. trad. 3 = Trad.

Source

Dykstra, et al. (2004)

References

- Examples in book: <http://stats.lse.ac.uk/bergsma/cmm/R%20files/NKPS.R>
- Bergsma, W. P., Croon, M. A., & Hagenaars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. New York: Springer.
- Dykstra, P. A. Kalmijn, M., Knijn, T. C. M., Komter, A. E., Liefboer, A. C., & Mulder, C. H. (2004). *Codebook of the Netherlands Kinship Panel Study: A multi-actor, multi-method, panel study on solidarity, in family relationships. Wave 1* (Tech. Rep. No. NKPS Working Paper 1). The Hague, The Netherlands: NICI.

Examples

```

data(NKPS)
attach(NKPS)

#####
# Chapter 2

# Table 2.8
Table.2.8 <- array(NA,c(4,4,4)); k <- 0
for (i in levels(P)) for (j in levels(C)){
  k <- k+1
  Table.2.8[, ,k] <- addmargins(t(table(NKPS[,c(3,4)] [C==j & P==i,])))
}
dimnames(Table.2.8) <- list(c(levels(PS),"Total"),c(levels(CS),"Total"),
                           c("Father-Son","Father-Daughter","Mother-Son","Mother-Daughter"))
Table.2.8

# Table 2.9
Table.2.9 <- cbind(table(PS),table(CS),table(c(CS[C=="Son"],PS[P=="Father"])),
                  table(c(CS[C=="Daughter"],PS[P=="Mother"])))
dimnames(Table.2.9)[[2]] <- c("Parent","Child","Men","Women")
addmargins(Table.2.9)[,-5]

# Table 2.10

# Table 2.11

#####
# Data
recAB = NKPS[,c(3,4)]
recPCAB = NKPS[,c(1,2,3,4)]
recA1A2B1B2 = NKPS[,c(3,4,5,6)]

# list of frequencies in table AB
nAB = c(t(ftable(recAB)))

#####
# table AB and GT

```

```
# at produces marginal distributions of A and B, or the 2x3 table GT;
# G = generation and T = attitude
at <- MarginalMatrix(c("A", "B"), list(c("A"), c("B")), c(3, 3));
bt <- ConstraintMatrix(c("G", "T"), list(c("G"), c("T")), c(2, 3));

model1 <- list(bt, "log", at);

nkps1 <- MarginalModelFit(dat = recAB, model = model1, ShowParameters = TRUE,
  Labels = list(list("G", c("men", "women")), "T"),
  CoefficientDimensions = c(2, 3), ShowProgress = 10)
```

SampleStatistics *SampleStatistics*

Description

Gives sample values, standard errors and z-scores of one or more coefficients. `SampleStatistics(dat, coeff)` gives exactly the same output as `ModelStatistics(dat, dat, "SaturatedModel", coeff)`.

Usage

```
SampleStatistics(dat, coeff, CoefficientDimensions = "Automatic",
  Labels = "Automatic", ShowCoefficients = TRUE, ParameterCoding = "Effect",
  ShowParameters = FALSE, ShowCorrelations = FALSE, Title = "", ShowSummary = TRUE)
```

Arguments

<code>dat</code>	observed data as a list of frequencies or as a data frame
<code>coeff</code>	list of coefficients, can be obtained using SpecifyCoefficient , or a predefined function such as "log"
<code>CoefficientDimensions</code>	numeric vector of dimensions of the table in which the coefficient vector is to be arranged
<code>Labels</code>	list of characters or numbers indicating labels for dimensions of table in which the coefficient vector is to be arranged
<code>ShowCoefficients</code>	boolean, indicating whether or not the coefficients are to be displayed
<code>ShowParameters</code>	boolean, indicating whether or not the parameters (computed from the coefficients) are to be displayed
<code>ParameterCoding</code>	Coding to be used for parameters, choice of "Effect", "Dummy" and "Polynomial"
<code>ShowCorrelations</code>	boolean, indicating whether or not to show the correlation matrix for the estimated coefficients
<code>Title</code>	Title of computation to appear at top of screen output
<code>ShowSummary</code>	Show summary on the screen

Details

The data can be a data frame or vector of frequencies. `MarginalModelFit` converts a data frame `dat` using `c(t(ftable(dat)))`.

For `ParameterCoding`, the default for "Dummy" is that the first cell in the table is the reference cell. Cell (i, j, k, \dots) can be made reference cell using `list("Dummy", c(i, j, k, \dots))`. For "Polynomial" the default is to use centralized scores based on equidistant (distance 1) linear scores, for example, if for $i = 1, 2, 3, 4$,

$$\mu_i = \alpha + q_i\beta + r_i\gamma + s_i\delta$$

where β is a quadratic, γ a cubic and δ a quartic effect, then q_i takes the values $(-1.5, -.5, .5, 1.5)$, r_i takes the values $(1, -1, -1, 1)$ (centralized squares of the q_i), and s_i takes the values $(-3.375, -.125, .125, 3.375)$ (cubes of the q_i).

Value

A subset of the output of `MarginalModelFit`.

Author(s)

W. P. Bergsma <w.p.bergsma@lse.ac.uk>

References

Bergsma, W. P. (1997). *Marginal models for categorical data*. Tilburg, The Netherlands: Tilburg University Press. http://stats.lse.ac.uk/bergsma/pdf/bergsma_phdthesis.pdf

Bergsma, W. P., Croon, M. A., & Hagenars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. Berlin: Springer.

See Also

[ModelStatistics](#), [MarginalModelFit](#)

Examples

```
## Not run:
data(BodySatisfaction)

## Table 2.6 in Bergsma, Croon and Hagenars (2009). Loglinear parameters for marginal table IS
## We provide two to obtain the parameters

dat <- BodySatisfaction[,2:8]      # omit first column corresponding to gender

# matrix producing 1-way marginals, ie the 7x5 table IS
at75 <- MarginalMatrix(var = c(1, 2, 3, 4, 5, 6, 7),
  marg = list(c(1),c(2),c(3), c(4),c(5),c(6),c(7)), dim = c(5, 5, 5, 5, 5, 5))

# First method: the "coefficients" are the log-probabilities, from which all the
# (loglinear) parameters are calculated
stats <- SampleStatistics(dat = dat, coeff = list("log",at75), CoefficientDimensions = c(7, 5),
  Labels = c("I", "S"), ShowCoefficients = FALSE, ShowParameters = TRUE)
```

```
# Second method: the "coefficients" are explicitly specified as being the
# (highest-order) loglinear parameters
loglinpar75 <- SpecifyCoefficient("LoglinearParameters", c(7, 5))
stats <- SampleStatistics(dat = dat, coeff = list(loglinpar75, at75),
  CoefficientDimensions = c(7,5), Labels = c("I","S"))

## End(Not run)
```

Smoking

Smoking cessation after experimental intervention

Description

Data from an experiment designed for the investigation of the effectiveness of a particular expert system intervention to convince people to quit smoking. $N = 4,144$ subjects were randomly assigned to either the control (assessment only) or the experimental condition (TTM intervention). Information was collected on their smoking habits and their attitudes towards smoking at the start of the study, at the sixth, twelfth, eighteenth, and twenty-fourth month. For more detailed information see Bergsma et al. (2009) and Prochaska et al. (2001).

The data are tabulated in Bergsma, Croon, and Hagenaaars (2009, Tables 5.11 to 5.14).

Section 5.2.3 in Bergsma, Croon and Hagenaaars (2009).

Usage

```
data(Smoking)
```

Format

A data frame with 4144 observations on the following variables.

Group (factor): 1 = TTM intervention; 2 = Assessment only.

smst00 Behavior at beginning (ordered): 1 = Precontemplation; 2 = Contemplation; 3 = Preparation; 4 = Action; 5 = Maintenance.

smst06 Behavior after 6 months (ordered): see smst00

smst12 Behavior after 12 months (ordered): see smst00

smst18 Behavior after 18 months (ordered): see smst00

smst24 Behavior after 24 months (ordered): see smst00

Source

Cancer Prevention Research Center, University of Rhode Island, US. See Prochaska, Velicer, Fave, Rossi & Tosh (2001).

References

Examples in book: <http://stats.lse.ac.uk/bergsma/cmm/R%20files/Smoking.R>

Bergsma, W. P., Croon, M. A., & Hagenaars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. New York: Springer.

Prochaska, J. O., Velicer, W. F., Fava, J. L. Rossi, J. S., & Tosh, J. Y. (2001). Evaluating a population-based recruitment approach and a stage-based expert system intervention for smoking cessation. *Addictive Behaviors*, 26, 583-602.

Examples

```
#####
# Read data
data(Smoking)

## Not run:

dat <- Smoking

#####
# Table TXBR
# matrix producing 4x2x3x6 table TXBR

atTXBR <- MarginalMatrix(var = c("X", "B", "R1", "R2", "R3", "R4"),
  marg = list(c("X", "B", "R1"), c("X", "B", "R2"), c("X", "B", "R3"), c("X", "B", "R4")),
  dim = c(2, 3, 5, 5, 5, 5))
bt <- ConstraintMatrix(var = c("T", "X", "B", "R"), suffconfigs = list(c("T", "X", "B"), c("R")),
  dim = c(4, 2, 3, 5))
model = list(bt, "log", atTXBR)

fit = MarginalModelFit(dat = dat, model = model, MaxStepSize = .3, MaxSteps = 100,
  ShowProgress = 5)

## End(Not run)
```

SpecifyCoefficient *SpecifyCoefficient*

Description

Gives the generalized exp-log specification for various coefficients

Usage

```
SpecifyCoefficient(name, arg = NULL, rep = 1, data = NULL)
```

Arguments

name	character: name of desired coefficient
arg	an argument specific to the coefficient, e.g., a vector of scores or number of rows and columns.
data	data set. Necessary for MEL estimation
rep	number of repetitions of the coefficient

Details

Currently the following coefficients are implemented:

```
SpecifyCoefficient("Mean",arg = scores)
SpecifyCoefficient("Variance", arg = scores)
SpecifyCoefficient("StandardDeviation", arg = scores)
SpecifyCoefficient("GiniMeanDifference", arg = scores)
SpecifyCoefficient("Entropy", arg = k)
SpecifyCoefficient("DiversityIndex", arg = k)
SpecifyCoefficient("Covariance",arg = list(xscores,yscores))
SpecifyCoefficient("Correlation",arg = list(xscores,yscores))
SpecifyCoefficient("KendallTau",arg = list(r,c))
SpecifyCoefficient("GoodmanKruskalGamma",arg = list(r,c))
SpecifyCoefficient("CohenKappa",r)
SpecifyCoefficient("CronbachAlpha",arg = list(items,item.scores), data = X)
SpecifyCoefficient("Hij")
SpecifyCoefficient("DifferenceInProportions",arg = m)
SpecifyCoefficient("LogOddsRatio")
SpecifyCoefficient("LoglinearParameters",arg = dim)
SpecifyCoefficient("Probabilities",arg = dim)
SpecifyCoefficient("LogProbabilities",arg = dim)
SpecifyCoefficient("ConditionalProbabilities",arg = list(var,condvar,dim))
SpecifyCoefficient("AllMokkenHj", arg = list(items,item.scores), data = X)
SpecifyCoefficient("MokkenH", arg = list(items,item.scores), data = X)
SpecifyCoefficient("OrdinalLocation-A", arg = arg)
SpecifyCoefficient("OrdinalLocation-L", arg = arg)
SpecifyCoefficient("OrdinalDispersion-A", arg = arg)
SpecifyCoefficient("OrdinalDispersion-L", arg = arg)
```

Here, scores is a score vector, e.g., $c(1,2,3,4,5)$; k is the number of cells in a table; r is the number of rows and columns of a square table; dim is the dimension of the table; items are the columns in the data matrix that should be used for computing the statistic; item.scores is the number of item scores (e.g., 2 for dichotomous items, or 5 for Likert items); X is the $N \times J$ data matrix. "LoglinearParameters" gives the highest order loglinear parameters (loglinear parameters can also be obtained as output of SampleStatistics, ModelStatistics or MarginalModelFit by setting ShowParameters=TRUE and the coefficients equal to log probabilities.

Value

output is of the form `list(matlist, funlist)` where `matlist` is a list of matrices and `funlist` is a list of function names, which can currently be "log", "exp", "identity", "xlogx" (i.e., $f(x) = x \log(x)$), "xbarx" (i.e., $f(x) = x(1 - x)$), "sqrt"

Author(s)

W. P. Bergsma <w.p.bergsma@lse.ac.uk>

References

Bergsma, W. P. (1997). *Marginal models for categorical data*. Tilburg, The Netherlands: Tilburg University Press. http://stats.lse.ac.uk/bergsma/pdf/bergsma_phdthesis.pdf

Bergsma, W. P., Croon, M. A., & Hagenaars, J. A. P. (2009). *Marginal models for dependent, clustered, and longitudinal categorical data*. Berlin: Springer.

See Also

[ConstraintMatrix](#), [DesignMatrix](#), [MarginalMatrix](#)

Examples

```
SpecifyCoefficient("Mean", arg = c(1,2,3))
SpecifyCoefficient("Variance", arg = c(1,2,3))
SpecifyCoefficient("StandardDeviation", arg = c(1,2,3))
SpecifyCoefficient("GiniMeanDifference", arg = c(1,2,3))
SpecifyCoefficient("Entropy", arg = 5)
SpecifyCoefficient("DiversityIndex", arg = 5)
SpecifyCoefficient("Covariance", arg = list(c(1,2,3), c(1,2,3)))
SpecifyCoefficient("Correlation", arg = list(c(1,2,3), c(1,2,3)))
SpecifyCoefficient("KendallTau", arg = list(3,4))
SpecifyCoefficient("GoodmanKruskalGamma", arg = list(3,4))
SpecifyCoefficient("CohenKappa", arg = 3)
SpecifyCoefficient("DifferenceInProportions", arg = 1)
SpecifyCoefficient("LogOddsRatio", arg = 1)
SpecifyCoefficient("LoglinearParameters", arg = c(3,3))
SpecifyCoefficient("Probabilities", arg = 8)
SpecifyCoefficient("LogProbabilities", arg = 8)
# conditional probabilities for 3x3 table, conditioning on first variable
SpecifyCoefficient("ConditionalProbabilities", arg = list(c(1,2), c(1), c(3,3)))
```

Description

Data set TestCronbachAlpha is a simulated data set that is used to demonstrate the statistical testing of three relevant hypotheses involving Cronbach's alpha: H01: alpha equals a particular criterion; H02: testing the equality of two alpha coefficients for independent samples; and H03: testing the equality of two alpha coefficients for dependent samples.

This R document file may be regarded as an appendix to Kuijpers, Van der Ark, and Croon (2012) who discussed this topic. Hence, all references to equations pertain to this paper. The *Details* section describes the required objects for testing the three hypotheses. The *Examples* section describes the actual code required for testing the three hypotheses.

Usage

```
data(TestCronbachAlpha)
```

Format

A 400 by 21 matrix, representing the dichotomous item scores of 400 respondents from two groups for two tests. The first column is the grouping variable: Group 1 and Group 2 each consist of 200 observations. Columns 2-11 are the items score of Test 1. Columns 12-21 are the item scores of Test 2. So each test includes $J = 10$ items having $K = 2$ item scores. Note that in Kuijpers et al. (2012), k is used rather than K ; $k = K - 1$. Data files TestCronbachAlphaH1 <- TestCronbachAlpha[1:200,2:11] TestCronbachAlphaH2 <- TestCronbachAlpha[1:400,1:11] and TestCronbachAlphaH3 <- TestCronbachAlpha[1:200,2:21] will be used to test hypotheses H01, H02, and H03, respectively.

Details

Vector \mathbf{m} is estimated under the general categorical marginal model $\mathbf{g}(\mathbf{m}) = \mathbf{d}$. Objects `coeff`, `bt`, and `at` define function $\mathbf{g}(\mathbf{m})$.

`coeff` Includes the design matrices and functions (i.e., `exp` and `log`) of the coefficients of interest. Function `SpecifyCoefficient` returns the design matrices and functions of several prespecified coefficients, including `C`. The argument `arg` in `SpecifyCoefficient` specifies for which of the J marginals Cronbach's alpha should be computed. Furthermore, the argument `data` in `SpecifyCoefficient` specifies for which data set Cronbach's alpha should be computed.

For hypothesis H01, which involves only one Cronbach's alpha, `coeff` is obtained by `coeff = SpecifyCoefficient(name = "CronbachAlpha", arg = list(list(1 : J), K), data = mydata)`. For H01, object `coeff` includes the design matrices and functions in Equation 10.

For hypothesis H02, which involves two alpha coefficients derived from two independent samples, `coeff` is obtained by `coeff = SpecifyCoefficient(name = "CronbachAlpha", arg = list(list(2 : (J + 1), 2 : (J + 1)), c(K, K), 1), data = mydata)`. For H02, `coeff` now includes the design matrices and functions in Equation 19.

For hypothesis H03, which involves two dependent alpha coefficients, `coeff` is obtained by `coeff = SpecifyCoefficient(name="CronbachAlpha", arg=list(list(test1, test2),c(K,K)), data=mydata,)`. For H03, object `coeff` includes the design matrices and functions in Equation 23.

`bt` Is called the *constraint matrix* and relates the coefficients defined in `coeff`. Hypothesis H01 pertains to one Cronbach's

- at Is called the *marginal matrix*. The marginal matrix was not specified for hypotheses H01 and H02, which is equivalent to the marginal matrix \mathbf{A}_0 (p. 16). For hypotheses H03 the marginal matrix is equal to design matrix \mathbf{A}_0 (p. 16). Function `MarginalMatrix` constructs the marginal matrix \mathbf{A}_0 .
- d Vector \mathbf{d} in Equation 9.

Function `MarginalModelFit` estimates the categorical marginal model (CMM), and requires the following arguments: the vector of observed frequencies, \mathbf{n} , and model specifications in `coeff`, `bt`, `at`, and `d`.

In the example for testing hypothesis H01, data set `TestCronbachAlphaH1` was used, which contained the 200 item-score vectors from the first group, for the first test. For this data set, Cronbach's alpha is equal to 0.793. If a researcher wants to test whether this value is significantly above .75, the software code for the first example in the paragraph `Examples` can be used (see below). First, the R package `cmm` needs to be invoked. Second, vector \mathbf{n} , the number of items J , the number of categories K , and criterion c in hypothesis H01 have to be defined. The fit of this marginal model is evaluated by G^2 , with $D = 1$ degree of freedom. In general, G^2 pertains to a two-sided test. However, here H01 is a one-sided hypothesis, the value of G^2 at the 2α level is used. For $\alpha = 0.05$, H01 must be rejected if $G^2 > 2.71$ (i.e., $p = .10$) and $r_alpha > c$. The results of the analysis show that $G^2 = 3.301$ with $p = 0.069$, so for this example we can conclude that the alpha of this data set (i.e., $r_alpha = 0.793$) is significantly above .75.

For testing hypothesis H02, data set `TestCronbachAlphaH2` was used, which contained the item-score vectors from the two independent groups for the first test, and an additional variable indicating group membership. For this data set, Cronbach's alpha for the first independent group is equal to 0.793, for the second independent group alpha is equal to 0.737. To test whether the alphas of the two independent groups are equal, the software code for the second example in the paragraph `Examples` can be used (see below). Note that the first item indicates group membership. Hence, for J items, vector \mathbf{n} is based on $J+1$ patterns. G^2 is used to assess the fit of this marginal model with $D = 1$ degree of freedom, so H02 must be rejected if $G^2 > 3.84$ (i.e., $\alpha = .05$). The results of the analysis show that $G^2 = 2.774$ with $p = 0.096$, so for this example we can conclude that the alphas of the two independent samples (i.e., $r_alpha_g1 = 0.793$ and $r_alpha_g2 = 0.737$) are equal.

For hypothesis H03, data set `TestCronbachAlphaH3` was used, which contained the 200 item-score vectors from the first group for the two tests. The data of each test forms one dependent group. For this data set, Cronbach's alpha for the first dependent group is equal to 0.793, for the second dependent group alpha is equal to 0.696. For H03, the marginal matrix is not implemented in the package as a code yet, so it has to be computed ad hoc. To test whether the alphas of the two dependent groups are equal, the software code for the third example in the paragraph `Examples` can be used (see below). G^2 is used to assess the fit of this marginal model with $D = 1$ degree of freedom. The results of the analysis show that $G^2 = 9.898$ with $p = 0.002$. Using $\alpha = .05$, we can conclude that the alphas of the two dependent samples (i.e., $r_alpha_t1 = 0.793$ and $r_alpha_t2 = 0.696$) are not equal to each other.

Author(s)

Renske E. Kuijpers, L. Andries van der Ark

References

Kuijpers, R. E., Van der Ark, L. A., & Croon, M. A. (2012). Testing hypotheses involving Cronbach's alpha using marginal models. Manuscript submitted for publication.

See Also

[cmm](#), [SpecifyCoefficient](#), [MarginalMatrix](#),

Examples

```
data(TestCronbachAlpha)

#Example 1: Testing H01.

# Invoke cmm
library(cmm)

# Data
TestCronbachAlphaH1 <- TestCronbachAlpha[1 : 200, 2 : 11]

# Transform data into vector of frequencies n
n <- as.matrix(table(apply(TestCronbachAlphaH1, 1, paste, collapse = "")))

# Specify number of items
J <- 10

# Specify number of item scores
K <- 2

# Specify criterion for Hypothesis H01
criterion <- .75

# Compute object coeff
coeff <- SpecifyCoefficient(name = "CronbachAlpha", arg = list(list(1 : J), K),
  data = TestCronbachAlphaH1)

# Compute object at (marginal matrix)
L <- ncol(coeff[[1]][[5]])
at <- diag(L)

# Compute object bt (constraint matrix)
bt <- matrix(1)

# Compute object d
d <- criterion

# Compute CMM
model <- list(bt, coeff, at, d)
fit <- MarginalModelFit(n, model, MaxError = 1e-04)

#Example 2: Testing H02.
```

```

# Data
TestCronbachAlphaH2 <- TestCronbachAlpha[1 : 400, 1 : 11]

# Transform data into vector of frequencies n
n <- as.matrix(table(apply(TestCronbachAlphaH2, 1, paste, collapse = "")))

# Specify number of items
J <- 10

# Specify number of item scores
K <- 2

# Compute object coeff
coeff <- SpecifyCoefficient(name = "CronbachAlpha", arg = list(list(2 : (J + 1),
  2 : (J + 1)), c(K, K), 1), data = TestCronbachAlphaH2,)

# Compute object at (marginal matrix)
L <- ncol(coeff[[1]][[5]])
at <- diag(L)

# Compute object bt (constraint matrix)
bt <- matrix(c(1,-1),1,2)

# Compute object d
d <- rep(0,nrow(bt))

# Compute CMM
model <- list(bt,coeff,at,d)
fit <- MarginalModelFit(n, model, MaxError = 1e-04)

#Example 3: Testing H03.

# Data
TestCronbachAlphaH3 <- TestCronbachAlpha[1 : 200, 2 : 21]

# Transform data into vector of frequencies n
n <- as.matrix(table(apply(TestCronbachAlphaH3, 1, paste, collapse = "")))

# Specify number of items
J <- 20

# Specify number of item scores
K <- 2

# Specify which items belong to which test
test1 <- 1 : 10
test2 <- 11 : 20

# Compute object coeff
coeff <- SpecifyCoefficient(name = "CronbachAlpha", arg = list(list(test1,
  test2), c(K, K)), data = TestCronbachAlphaH3,)

# Compute object at (marginal matrix)

```

```
x <- dimnames(n)[[1]]
p1 <- sort(unique(substr(x, test1[1] ,test1[length(test1)])))
p2 <- sort(unique(substr(x, test2[1] ,test2[length(test2)])))
U1 <- matrix(NA, length(p1), length(x))
for (h1 in 1 : length(p1))
U1[h1, ] <- as.numeric(substr(x, test1[1], test1[length(test1)]) == p1[h1])
U2 <- matrix(NA, length(p2), length(x))
for (h2 in 1 : length(p2))
U2[h2, ] <- as.numeric(substr(x, test2[1], test2[length(test2)]) == p2[h2])
at <- rbind(U1, U2)

# Compute object bt (constraint matrix)
bt <- matrix(c(1, -1), 1, 2)

# Compute object d
d <- rep(0, nrow(bt))

# Compute CMM
model <- list(bt, coeff, at, d)
fit <- MarginalModelFit(n, model, MaxError = 1e-04)
```


Index

*Topic **datasets**

Antisemitism, 3
BodySatisfaction, 4
ClarenceThomas, 6
DutchConcern, 14
DutchPolitics, 15
ErieCounty, 17
EVS, 18
GSS93, 19
LaborParticipation, 23
MarihuanaAlcohol, 30
NES, 34
NKPS, 35
Smoking, 39
TestCronbachAlpha, 42

*Topic **package**

cmm-package, 2

*Topic **univar**

ConstraintMatrix, 8
DesignMatrix, 10
DirectSum, 13
JoinModels, 21
MarginalMatrix, 24
MarginalModelFit, 26
ModelStatistics, 32
SampleStatistics, 37
SpecifyCoefficient, 40

DutchConcern, 2, 14
DutchPolitics, 2, 15

ErieCounty, 2, 17
EVS, 2, 18

GSS93, 2, 19

JoinModels, 2, 21

LaborParticipation, 2, 23

MarginalMatrix, 2, 9, 11, 24, 42, 44, 45
MarginalModelFit, 2, 22, 26, 33, 38, 44
MarihuanaAlcohol, 2, 30
ModelStatistics, 2, 27, 30, 32, 33, 38

NES, 2, 34

NKPS, 2, 35

nkps1 (NKPS), 35

NKPS2, 2

NKPS2 (NKPS), 35

SampleStatistics, 2, 30, 37

Smoking, 2, 39

SpecifyCoefficient, 2, 22, 32, 37, 40, 43, 45

TestCronbachAlpha, 42

Antisemitism, 2, 3

BodySatisfaction, 2, 4

ClarenceThomas, 2, 6

cmm, 45

cmm (cmm-package), 2

cmm-package, 2

ConstraintMatrix, 2, 8, 9, 11, 13, 25, 42

DesignMatrix, 2, 9, 10, 13, 25, 42

DirectSum, 2, 9, 11, 13, 13, 22, 25