Package ‘catspec’

January 2, 2012

Version 0.95

Title Special models for categorical variables

Author John Hendrickx <John_Hendrickx@yahoo.com>

Maintainer John Hendrickx <John_Hendrickx@yahoo.com>

Description ‘sqtab’ contains a set of functions for estimating loglinear models for square tables such as quasi-independence,symmetry, uniform association. ‘mclgen’ restructures a dataframe to enable the estimation of a multinomial logistic model using the conditional logit program ‘clogit’. This allows greater flexibility in imposing constraints on the response variable. One application is to specify aforementioned models for square tables as multinomial logistic models with covariates at the respondent level. ‘ctab’ simplifies the production of (multiway(percentage tables.

License GPL (>= 2)

URL http://home.wanadoo.nl/john.hendrickx/statres/

Repository CRAN

Date/Publication 2010-06-03 13:49:33

R topics documented:

ctab *Percentage tables*

Description

Produces one-way, two-way or multi-way percentage tables

Usage

```
ctab(...,dec.places=NULL,digits=NULL,type=NULL,style=NULL,row.vars=NULL,col.vars=NULL,percentages=N
```

Usage

```
ctab(..., dec.places=2, digits=dec.places, type=c("n", "row", "column", "total"),
style="long", row.vars=NULL, col.vars=NULL, percentages=TRUE, addmargins=FALSE)

## S3 method for class 'ctab'
print(x, dec.places=x$dec.places, addmargins=x$addmargins, ...)

## S3 method for class 'ctab'
summary(object,...)
```

Arguments

`...`	either • one or more factors, • a class `table` • a class `ftable` table object, • or a class `ctab` table object
`dec.places`	number of decimal places (default 2)
`digits`	synonym for `dec.places`, for compatability with previous version
`type`	Row, column, total percentages or counts (`type= n`). Multiple values may be specified as a character vector. Partial matchin is used.
`style`	Applicable if more than one percentage type is specified. If `style=long`, percentages are printed underneath each other. If `style=wide`, the percentages are printed side by side
`row.vars`	Same as `ftable`: *"a vector of integers giving the numbers of the variables, or a character vector giving the names of the variables to be used for the rows of the [] table"*
`col.vars`	*"a vector of integers giving the numbers of the variables, or a character vector giving the names of the variables to be used for the columns of the [] table"*
`percentages`	If `FALSE`, proportions rather than percentages are printed
`addmargins`	Use `addmargins=TRUE` to add subtotals to the table
`x`	is a tables object created by `ctab`
`object`	is a tables object created by `ctab`

Details

ctab uses ftable and prop.table to produce one-way frequency tables, two-way crosstables, or multi-way percentage tables. More than one percentage type may be specified, in which case "percentage type" is an unnamed dimension of the table. row.vars and col.vars can be used to control the layout of multi-way tables using the facilities of ftable. Subtotals can be added by specifiying addmargins=TRUE. Note that [gmodels]CrossTable in the gmodels package also provides an easy method for producing percentage tables, but only for two-way tables.

If ctab is specified with no further options and for more than one factor, the output is identical to that of ftable. If a single factor is specified, the default is to print the frequencies column-wise with the percentaqes next to them.

Value

An object of class "ctab". print.ctab prints the table, summary.ctab passes the frequency table on to summary.table, which prints the number of cases, number of factors, and a chi-square test of independence.

table	A class(table) object containing the table counts. Used by summary.ctab and by ctab itself if a ctab object is used as input.
ctab	A class(ftable) object containing the percentage types specified. This is printed by print.ctab.
row.vars	The row.vars options as numeric vectors
col.vars	The col.vars options as numeric vectors
dec.places	The dec.places option
type	The type option
style	The style option
percentages	The percentages option
addmargins	The addmargins option

Author(s)

John Hendrickx <<John_Hendrickx@yahoo.com>>

References

http://www.xs4all.nl/~jhckx/

See Also

table, ftable, addmargins, prop.table, xtabs, [gmodels]CrossTable

Examples

```
ftable(Titanic)
ctab(Titanic) # same output
ctab(Titanic,type="r")
ctab(Titanic,type=c("n","r"),addmargins=TRUE)
ctab(Titanic,type=c("n","c","t","r"),style="w")
mytab<-ftable(Titanic,row.vars=c(1,3),type="r")
mytab
ctab(mytab)
newtab<-ctab(mytab,type="r")
newtab
summary(newtab)

#second example using a data frame rather than table data
data(logan)
class(logan) #"data.frame"
ctab(occ)
ctab(occ,addmargins=TRUE)
ctab(occ,style="w",type="c")
ctab(occ,style="l",type="n")
z<-ctab(occ,addmargins=TRUE,style="l")
z
print(z,addmargins=FALSE,dec.places=5)
summary(z)

t<-ctab(focc,occ,type=c("n","r","c"))
t
summary(t)
```

FHtab *Intergenerational mobility table*

Description

The FHtab data frame has 25 rows and 3 columns. OccFather is father's occupation, OccSon is son's occupation, both have 5 categories:

- Upper nonmanual
- Lower nonmanual
- Upper manual
- Lower manual
- Farm

Freq contains the table frequencies

Usage

```
data(FHtab)
```

Source

Table from page 11 of *Mobility Tables*. Original source: page 49 of *Opportunity and Change*.

References

Hout, Michael. (1983). *Mobility Tables.* Sage Publication 07-03

Featherman, David L., Robert M. Hauser. (1978). *Opportunity and change.* New York: Academic Press.

`logan` *Data from the 1972-78 GSS data used by Logan (1983: 332-333)*

Description

Intergenerational occupational mobility data with covariates.

Usage

```
data(logan)
```

Format

A data frame with 838 observations on the following 4 variables.

occ a factor with levels `farm`, `operatives`, `craftsmen`, `sales`, and `professional`

focc a factor with levels `farm`, `operatives`, `craftsmen`, `sales`, and `professional`

educ a numeric vector

black a factor with levels `non-black` and `black`

Source

1972-78 GSS (cf. Logan, 1983: 332-333)

References

Logan, John A. (1983). A Multivariate Model for Mobility Tables. *American Journal of Sociology* 89: 324-349.

mclgen *Restructure a data-frame as a person-choice file*

Description

mclgen restructures a data-frame into a *person-choice* file for estimation of a multinomial logit as a conditional logit model

Usage

```
mclgen(datamat, catvar)
```

Arguments

datamat	A data-frame to be transformed into a *person-choice* file
catvar	A factor representing the response variable (i.e. the dependent variable in a multinomial logistic model)

Details

A multinomial logit model can be estimated using a program for conditional logit regression. This will produce the same coefficients and standard errors but allows greater flexibility for imposing restrictions on the dependent variable.

To estimate the multinomial logistic model as a conditional logit model, the data must be restructured as a *person-choice* file. mclgen performs this operation such that:

- each record of the data-frame is duplicated *ncat* times, where *ncat* is the number of categories of the response variable
- A new variable id is created to index respondents. This variable is used as the stratifying variable in clogit
- A new variable newy is created to index response options for each respondent
- A new variable depvar is created which is equal to 1 for the record corresponding with the respondent's actual choice and is 0 otherwise

depvar is the dependent variable in clogit. The main effects of catvar correspond with the intercept term of a multinomial logit model, interactions of catvar with predictor variables correspond with the effects of these variables in a multinomial logit model.

Since catvar is now on the right-hand side of the model equation, restrictions can be imposed in the usual fashion. For example, by using [MASS]contr.sdif in the MASS package for catvar, an adjacent logit model is obtained (Agresti 1990: 318). By adding the dummy variables for two categories of catvar, an equality constraint can be imposed on those categories. These equality constraints can then be imposed on the effects of some predictor variables but not others.

Another use is to include a mobility model in a multinomial logistic regression model. Mobility models are loglinear models for square tables. They lie in the space between a model of independence and a saturated model. This is accomplished by imposing restrictions on the interaction effect of the row and column variable. A number of these special models have been developed, see Hout (1983) or Goodman (1984) for an overview.

These loglinear mobility models can be seen as multinomial logistic regression models with special restrictions on the dependent variable. The nature of the restriction depends on the category of the predictor variable. In practise, mobility models can be included in an MCL model using the same specification as for a loglinear model. R functions for several common mobility models can be found in `sqtab`.

Value

A data-frame is returned, restructured as a *person-choice* file.

Note

The effects of predictor variables are modelled as interactions with `depvar` but the main effects of the predictor variables are not included in the model since these are constant within strata. This causes problems due to the way R handles contrasts in interaction effects. If lower order effects of variables are not included in a model then contrasts are not used in interactions with these variables (see `terms.object` under the heading `factors`). For example, in the model `~Y+Y:X1+Y:X2`, contrasts will only be used for the main effect of `Y`. For the interaction effects `Y:X1` and `Y:X2`, dummies for all levels of `Y` will be used. The dummies for `Y` are linearly dependent within strata of the conditional logit model, therefore a warning issues that *"X matrix deemed to be singular"* and the dummy for the last level of `Y` is dropped. So the model is estimated but `depvar` is not coded as intended.

The simplest workaround is to include the main effects of the predictor variables in the model, i.e. use `~Y*X1+Y*X2`. The main effects of `X1` and `X2` are constant within strata so a warning will issue that *"X matrix deemed to be singular"* and these effects will be dropped from the model. This has no further consequences and `depvar` is coded as intended.

A second workaround can be to use `model.matrix` to create the dummies for `depvar` using the intended contrast. See the example below.

`clogit` does not accept weights. A workaround is to call `coxph` directly (see example below).

Author(s)

John Hendrickx `<John\_Hendrickx@yahoo.com>`

References

Agresti, Alan. (1990). *Categorical data analysis*. New York: John Wiley & Sons.

Allison, Paul D. and Nicholas Christakis. (1994). Logit models for sets of ranked items. Pp. 199-228 in *Peter V. Marsden (ed.), Sociological Methodology*. Oxford: Basil Blackwell.

Breen, Richard. (1994). Individual Level Models for Mobility Tables and Other Cross- Classifications. *Sociological Methods & Research* 33: 147-173.

Goodman, Leo A. (1984). *The analysis of cross-classified data having ordered categories*. Cambridge, Mass.: Harvard University Press.

Hendrickx, John. (2000). Special restrictions in multinomial logistic regression. *Stata Technical Bulletin* 56: 18-26.

Hendrickx, John, Ganzeboom, Harry B.G. (1998). Occupational Status Attainment in the Netherlands, 1920-1990. A Multinomial Logistic Analysis. *European Sociological Review* 14: 387-403.

Hout, Michael. (1983). *Mobility Tables*. Sage Publication 07-031.

Logan, John A. (1983). A Multivariate Model for Mobility Tables. *American Journal of Sociology 89*: 324-349.

http://www.xs4all.nl/~jhckx/

See Also

[survival]clogit, [survival]coxph, [nnet]multinom, sqtab

Examples

```
## Example 1
# data from the Data from the 1972-78 GSS used by Logan (1983)
data(logan)

# create the "person-choice" file
pc<-mclgen(logan,occ)
summary(pc)
attach(pc)

library(survival)
# The following specification will work but R won't drop
# cl.lr<-clogit(depvar~occ+occ:educ+occ:black+strata(id),data=pc)
# However, R won't drop the first category of "occ"
# in the interaction effects. The last category will be omitted
# instead due to linear dependence within strata.
# Fix for the problem, create dummies manually for "occ"
occ.X<-model.matrix(~pc$occ)
occ.X<-occ.X[,attributes(occ.X)$assign==1]
cl.lr<-clogit(depvar~occ.X+occ.X:educ+occ.X:black+strata(id),data=pc)
summary(cl.lr)

# Estimate a "quasi-uniform association" loglinear model for "focc" and "occ"
# with "educ" and "black" as covariates at the respondent level
cl.qu<-clogit(depvar~occ.X+occ.X:educ+occ.X:black+
  mob.qi(focc,occ)+mob.unif(focc,occ)+strata(id),data=pc)
summary(cl.qu)

data(housing,package="MASS")
housing.prsch<-mclgen(housing,Sat)
library(survival)
# clogit doesn't support the weights argument at present
# a work-around is to call coxph directly
# coxph warns that X is singular, because the main
# effects of Infl, Type, and Cont are dropped
coxph.prsch<-coxph(Surv(rep(1, NROW(housing.prsch)), depvar) ~
  Sat+Sat*Infl+Sat*Type+Sat*Cont+strata(id),
  weights = housing.prsch$Freq, data = housing.prsch)
summary(coxph.prsch)
```

```
# the same model using multinomial logistic regression
library(nnet)
house.mult<- multinom(Sat ~ Infl + Type + Cont, weights = Freq,
                      data = housing)
summary(house.mult,correlation=FALSE)

# compare the coefficients
m1<-coef(coxph.prsch)
m1<-m1[!is.na(m1)]
dim(m1)<-c(2,7)
m2<-coef(house.mult)
m1
m2
m1-m2
max(abs(m1-m2))
mean(abs(m1-m2))
```

---

sqtab *sqtab: models for square tables*

Description

These functions are used to estimate loglinear models for square tables such as quasi-independence, quasi-symmetry.

Usage

```
mob.qi(rowvar, colvar, constrained = FALSE, print.labels = FALSE)
mob.eqmain(rowvar, colvar, print.labels = FALSE)
mob.symint(rowvar, colvar, print.labels = FALSE)
mob.cp(rowvar,colvar)
mob.unif(rowvar, colvar)
mob.rc1(rowvar, colvar, equal = FALSE, print.labels = FALSE)
fitmacro(object)
check.square(rowvar, colvar, equal = TRUE)
```

Arguments

`rowvar`	Factor representing the row variable
`colvar`	Factor representing the column variable
`print.labels`	If FALSE (default) then numeric values rather than factor values are printed for compact results
`equal`	(`mob.rc1`) If TRUE, a homogeneous row and column effects model 1 with equal scale values for the row and column variables is estimated. Otherwise, a (regular) RC1 model is estimated with different scale values for the row and column variables (`check.square`) If TRUE, the row and column variables must have the same number of categories

`constrained`	(`mob.qi`)If TRUE, a quasi-independence model-constrained is estimated with a single parameter for the diagonal cells
`object`	(`fitmacro`) An object of class `glm` for `family=poisson` and `link=log`

Details

These functions are used to estimate loglinear models for square tables:

mob.qi Quasi-independence

mob.eqmain Equal main effects (Hope's halfway model)

mob.symint Symmetric interaction

mob.cp Crossings-parameter model

mob.unif Uniform association

mob.rc1 Row and columns model 1

fitmacro Calculates `BIC` and `AIC` relative to a saturated loglinear model

check.square Internal function to check if row and column variables are both factors with the same number of levels

Value

`mob.qi`	A factor that will produce coefficients for the diagonal cells of a table, using off-diagonal cells as base category
`mob.eqmain`	A design matrix with equality contraints on the main effects
`mob.symint`	A design matrix for an interaction with equality contraints on coefficients on opposite sides of the diagonal
`mob.cp`	A set of vectors for a crossings-parameter model
`mob.unif`	A vector for a uniform association model
`mob.rc1`	A set of vectors for a row and columns model 1
`fitmacro`	Prints deviance, df, BIC, AIC, number of parameters and N
`check.square`	Stops function if either the row or column variable is not a factor or if the number of levels is unequal

Author(s)

John Hendrickx <`<John\_Hendrickx@yahoo.com>`>

References

Hout, Michael. (1983). *Mobility Tables*. Sage Publication 07-031.

Goodman, Leo A. (1984). *The analysis of cross-classified data having ordered categories.* Cambridge, Mass.: Harvard University Press.

See Also

`glm`, `mclgen`

Examples

```
# Examples of loglinear models for square tables,
# from Hout, M. (1983). "Mobility Tables". Sage Publication 07-031

# Table from page 11 of "Mobility Tables"
# Original source: Featherman D.L., R.M. Hauser. (1978) "Opportunity and Change."
# New York: Academic, page 49

data(FHtab)
FHtab<-as.data.frame(FHtab)
attach(FHtab)

xtabs(Freq ~ .,FHtab)

# independence model
indep<-glm(Freq~OccFather+OccSon,family=poisson())
summary(indep)
fitmacro(indep)

wt <- as.numeric(OccFather != OccSon)
qi0<-glm(Freq~OccFather+OccSon,weights=wt,family=poisson())
# A quasi-independence loglinear model, using structural zeros
# (page 23 of "Mobility Tables").
#  0  1  1  1  1   values of variable "wt"
#  1  0  1  1  1
#  1  1  0  1  1
#  1  1  1  0  1
#  1  1  1  1  0
qi0<-glm(Freq~OccFather+OccSon,weights=wt,family=poisson())
summary(qi0)
fitmacro(qi0)

# Quasi-independence using a "dummy factor" to create the design
# vectors for the diagonal cells (page 23).
#  1  0  0  0  0
#  0  2  0  0  0
#  0  0  3  0  0
#  0  0  0  4  0
#  0  0  0  0  5
glm.qi<-glm(Freq~OccFather+OccSon+mob.qi(OccFather,OccSon),family=poisson())
summary(glm.qi)
fitmacro(glm.qi)

# Quasi-independence constrained (QPM-C, page 31)
# Single immobility parameter
#  1  0  0  0  0
#  0  1  0  0  0
#  0  0  1  0  0
#  0  0  0  1  0
#  0  0  0  0  1
glm.q0<-glm(Freq~OccFather+OccSon+mob.qi(OccFather,OccSon,constrained=TRUE),family=poisson())
# slightly different results than Hout also found in Stata: L2=2567.658, q0=0.964
```

```
summary(glm.q0)
fitmacro(glm.q0)

# Quasi-symmetry using the symmetric cross-classification (page 23)
#  0  1  2  3  4   values of variable "sym"
#  1  0  5  6  7
#  2  5  0  8  9
#  3  6  8  0 10
#  4  7  9 10  0  */
glm.qsym<-
glm(Freq~OccFather+OccSon+mob.symint(OccFather,OccSon),family=poisson())
summary(glm.qsym)
fitmacro(glm.qsym)

symmetry<-glm(Freq~mob.eqmain(OccFather,OccSon)
+mob.symint(OccFather,OccSon),family=poisson())
summary(symmetry)
fitmacro(symmetry)

# Crossings parameter model (page 35)
#  0  v1 v1 v1 v1 |  0  0  v2 v2 v2 |  0  0  0  v3 v3 |  0  0  0  0 v4
#  v1 0  0  0  0  |  0  0  v2 v2 v2 |  0  0  0  v3 v3 |  0  0  0  0 v4
#  v1 0  0  0  0  |  v2 v2 0  0  0  |  0  0  0  v3 v3 |  0  0  0  0 v4
#  v1 0  0  0  0  |  v2 v2 0  0  0  |  v3 v3 v3 0  0  |  0  0  0  0 v4
#  v1 0  0  0  0  |  v2 v2 0  0  0  |  v3 v3 v3 0  0  |  v4 v4 v4 v4 0
glm.cp<-glm(Freq~OccFather+OccSon+mob.cp(OccFather,OccSon),family=poisson())
summary(glm.cp)
fitmacro(glm.cp)

# Uniform association model: linear by linear association (page 58)
glm.unif<-glm(Freq~OccFather+OccSon+mob.unif(OccFather,OccSon),family=poisson())
summary(glm.unif)
fitmacro(glm.unif)

# RC model 1 (unequal row and column effects, page 58)
# Fits a uniform association parameter and row and column effect
# parameters. Row and column effect parameters have the
# restriction that the first and last categories are zero.
glm.rc1<-glm(Freq~OccFather+OccSon+mob.rc1(OccFather,OccSon),family=poisson())
summary(glm.rc1)
fitmacro(glm.rc1)

# Homogeneous row and column effects model 1 (page 58)
# An equality restriction is placed on the row and column effects
glm.hrc1<-glm(Freq~OccFather+OccSon+mob.rc1(OccFather,OccSon,equal=TRUE),family=poisson())
# Results differ from those in Hout, replicated by other programs
summary(glm.hrc1)
fitmacro(glm.hrc1)
```

Index