Package ‘LabApplStat’

November 29, 2021

Title  Miscellaneous Scripts from the Data Science Laboratory (UCPH)
Version  1.4.3
Depends  stats, graphics, emmeans, ggplot2
Imports  ggraph, grid
Suggests  isdals, estimability, dobsn, tidyverse
Description  Miscellaneous scripts, e.g. functionality to make and plot factor diagrams for the statistical design.
License  GPL-3
Encoding  UTF-8
RoxygenNote  7.1.2
NeedsCompilation  no
Author  Bo Markussen [aut, cre] (<https://orcid.org/0000-0001-7793-9620>)
Maintainer  Bo Markussen <bomar@math.ku.dk>
Repository  CRAN
Date/Publication  2021-11-29 09:40:06 UTC

R topics documented:

chisq.test.simulate  .....................................................  2
DD  .................................................................  3
designDiagram-class  ....................................................  4
emmeans_ED  ...........................................................  7
minimum  ...............................................................  9
power.chisq.test.simulate  .......................................... 10

Index  12
chisq.test.simulate  Simulate Chi-squared tests with conditioning

Description

chisq.test.simulate simulates the chi-squared test for a 2-way contingency table.

Usage

chisq.test.simulate(x, conditioning = "total", x0 = NULL, B = 10000)

Arguments

x  matrix with the contingency table
conditioning character string specifying the simulation scenario. Defaults to "total". Other possible scenarios are "row", "col", and "both".
x0  matrix specifying the null distribution. Defaults to NULL, in which case the null is estimated from the observed data x.
B  integer specifying the number of replicates used in the Monte Carlo test. Defaults to 10000.

Details

Using conditioning="both" corresponds to selecting simulate.p.value=TRUE in chisq.test. However, conditioning on both row and column marginals appears to be rarely justified in real data. Instead conditioning="total" is the correct choice for testing independence. Similarly, conditioning="row" is recommended when the row marginals e.g. are fixed by experimental design. The option x0 has no effect when conditioning on both row and column marginals.

Value

An object of class "htest".

Note

The code has not been optimized for speed, and might be slow.

Author(s)

Bo Markussen

See Also

chisq.test
Examples

# The Avadex dataset
Xobs <- matrix(c(2,3,6,40),2,2)
rownames(Xobs) <- c("Avadex +","Avadex -")
colnames(Xobs) <- c("Tumor +","Tumor -")

# In this example only the rows appear to be fixed by experimental design.
# As is seen below, conditioning also on the columns is misleading conservative.
chisq.test.simulate(Xobs,"both")
chisq.test.simulate(Xobs,"row")
chisq.test.simulate(Xobs,"total")

# Conditioning both on row and column marginals is similar to chisq.test().
chisq.test(Xobs,simulate.p.value=TRUE)

## Design diagram for a linear model

**Description**

`DD` computes the Design Diagram for a linear model.

**Usage**

```r
DD(fixed, random = NULL, data, keep = ~1, center = FALSE, eps = 1e-12)
```

**Arguments**

- `fixed`: formula with fixed effects. A response may the specified, but this optional.
- `random`: formula with random effects. Defaults to NULL meaning that there are no other random effects than the residual, which is added to all designs.
- `data`: data frame with the explanatory variables and the response (if specified).
- `keep`: formula which effects that will not be removed in the collinarity analysis. Defaults to ~1 meaning that the intercept will be kept if it is present.
- `center`: boolean deciding whether to centralize numerical predictors when an intercept is present. Defaults to FALSE.
- `eps`: threshold for deeming singular values to be "zero". Defaults to 1e-12.

**Value**

An object of class `designDiagram-class`

**Author(s)**

Bo Markussen
See Also

minimum, plot.designDiagram

Examples

```r
# 3-way ANOVA
x <- factor(rep(rep(1:4, times=4), times=4))
y <- factor(rep(rep(1:4, times=4), each=4))
z <- factor(rep(rep(1:4, each=4), each=4))
myDD <- DD(~x+y*z, data=data.frame(x=x, y=y, z=z))
summary(myDD)

# Making the factor diagram closed under minima
mydata <- data.frame(age=rep(c("boy", "girl", "adult", "adult"), 4),
gender=rep(c("child", "child", "man", "woman"), 4))
myDD <- DD(~0+age+gender, data=mydata)
plot(myDD)

# Example of collinearity
mydata <- data.frame(age=rnorm(102), edu=rnorm(102), sex=factor(rep(c(1, 2), 51)))
mydata <- transform(mydata, exper=age-edu+0.1*rnorm(102))
mydata <- transform(mydata, wage=2*edu+2*exper+rnorm(102))
summary(myDD <- DD(wage~sex*(age+exper+edu), data=mydata))

# Growth of rats
antibiotica <- factor(rep(c(0, 40), each=6))
vitamin <- factor(rep(rep(c(0, 5), each=3), 2))
growth <- c(1.30, 1.19, 1.08, 1.26, 1.21, 1.19, 1.05, 1.00, 1.05, 1.05, 1.56, 1.55)
mydata <- data.frame(antibiotica=antibiotica, vitamin=vitamin, growth=growth)
myDD <- DD(growth~antibiotica*vitamin, data=mydata)
plot(myDD, "MSS")
plot(myDD, "I2")

# ANCOVA: Non-orthogonal design
library(isdals)
data(birthweight)
plot(DD(weight~sex*I(age-42), data=birthweight), "MSS")
plot(DD(weight~I(age-42)+sex:I(age-42)+sex, data=birthweight), "MSS")
```

---

**designDiagram-class**  
*The designDiagram class and some basic methods*

**Description**

Objects of class designDiagram as generated by `DD` is a list with entries as specified below.

- `response` Logical stating whether a response variable was present.
terms  Named vector with all terms in the design.
random.terms  Vector with the random terms in the design.
relations  Named matrix with relations between variables with the following interpretation: "0"=linear independent, "<"=row term is a subspace of column, "<:"=row term is a subspace of column term and no other terms are in between, ">" and ">:" the similar interpretation between columns and rows, name=name of minimum between row and column term.
inner  Named matrix of squared inner products of subspaces with nesting subspaces removed. Rounded at order of eps in the call to link{DD}. Used to decide orthogonality of the design.
N parm  Named vector with the number of parameters for the terms.
df  Named vector with the degrees of freedom for the terms.
SS  Named matrix with Sum-of-Squares if a response variable was specified.
MSS  Named matrix with Mean-Sum-of-Squares if a response variable was specified.
pvalue  Named matrix with p-values for Type-I F-tests. p-values are stated at the collapsed nesting, but F-test are done against the most coarse nested random effect.
sigma2  Named vector of random effects variance estimates.
varcov  Named list of variance-covariance matrix for fixed effects relative to each of the random effects. Rounded at order of eps.
coordinates  Data frame with node coordinates of the terms. Initialized in Sugiyama layout.

Usage

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

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

## S3 method for class 'designDiagram'
update(object, ...)

## S3 method for class 'designDiagram'
plot(
x, circle = "none", pvalue = (circle == "MSS"), sigma2 = NULL, kill = ~1, ca = FALSE, max.area = NULL, relative = 0.01, color = NULL, circle.scaling = 1, arrow.type = arrow(angle = 20, length = unit(4, "mm")), xlim = c(0, 1), ylim = c(0, 1), ...)
horizontal = TRUE,

Arguments

x object of class designDiagram

... not used.

object object of class designDiagram

circle character specifying which circles to draw at the terms: "none"=no circles, "SS"=a circle with area proportional to the associated Sum-of-Squares, "MSS"=a circle with area proportional to the associated Mean-Sum-of-Squares, "I"=a circle with area proportional to average information, "I2"=a circle with area proportional to average information of the parameter contrasts. Defaults to "none".

pvalue boolean specifying whether p-values should be inserted on the graphs. This is only possible if a response variable was specified. Defaults to TRUE is circle="MSS" and FALSE otherwise.

sigma2 vector of random effects variances. Defaults to NULL, in which case the estimates are used (if present), otherwise all variances are set to 1.

kill formula specifying which circles not to plot. Defaults to ~1 corresponding to not plotting the intercept term (that otherwise may overweight the remaining terms).

circle.scaling numeric specifying size scaling of circles. Defaults to 1, which corresponds to the largest circle having a radius that is half of the shortest distance between two nodes.

arrow.type specifying arrow heads via arrow. Defaults to arrow(angle=20,length=unit(4,"mm")).

xlim x-range of diagram plot. Defaults to c(0,1).

ylim y-range of diagram plot. Defaults to c(0,1).

horizontal boolean specifying if the design diagram should be drawn horizontally or vertically. Defaults to TRUE.

Details

For plot.designDiagram the options circle="SS" and circle="MSS" are only available if a response variable was specified for the design. For circle="I" and circle="I2" the color of the
circles visualize the coefficient of variation of the informations. For the computation of the informations the variances of the random effects are either estimated (if a response variable is present), all set to 1 (otherwise), or given via the option sigma2.

If color=\texttt{NULL} and \texttt{ca=FALSE}, then the defaults colors are "lightgreen" for Sum-of-Squares, "lightblue" for Mean-Sum-of-Squares, and a gradient from "limegreen" to "orange" for information spread. To specify a different color gradient in the latter case, then give a vector of two colors.

For update.designDiagram the second argument should be a data frame with new coordinates. This can be usefull for manually setting the coordinates for plotting.

\textbf{See Also}

\texttt{DD}

---

\textbf{emmeans\_ED} \hspace{1cm} \textit{Make emmeans object for an expected dose}

\textbf{Description}

Solves linear equations in continuous explanatory variables in order to find the expected dose. A typical application could be to find LD50, i.e. the lethal dose killing 50 percent of the population, from a probit analysis fitted by \texttt{glm}. The associated variance-covariance matrix is found using the Delta method.

\textbf{Usage}

\begin{verbatim}
emmeans_ED(
  object,
  specs = ~0,
  left = NULL,
  right = NULL,
  tran = NULL,
  p = 0.5,
  p.name = "probability"
)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item \textbf{object} \hspace{1cm} An object that can be given to \texttt{emmeans}. Typically a model fitted by \texttt{glm}.
  \item \textbf{specs} \hspace{1cm} As for \texttt{emmeans}. Typically as one-sided \texttt{formula}. Defaults to \texttt{~0}.
  \item \textbf{left} \hspace{1cm} A list specifying the left end point of the linear span of continuous variables in which to measure the ED values. Defaults to \texttt{NULL}.
  \item \textbf{right} \hspace{1cm} A list specifying the right end point of the linear span of continuous variables in which to measure the ED values. Defaults to \texttt{NULL}.
\end{itemize}
Possible transformation of the scale of the ED values. If given then backtrans-formation can be done using the technology of the emmeans. The default value tran=NULL corresponds to no transformation.

p  Numeric vector given the targeted predictions. Typically probabilities, where the default value p=0.5 corresponds to ED50.

p.name  The name of the variable containing p. If p contains more than one value, then this will also appear in misc$by.vars in the emmGrid object. Defaults to p.name="probability".

Find the 'expected dose' along a gradient in the space of numeric predictor variables. The options 'left' and 'right' specify the endpoints of this gradient. Typically these endpoints should be chosen as 0 and 1 for the numeric predictor of interest. If both endpoints are chosen as NULL then these choices are taken for all numeric predictors.

An object of class emmGrid-class.

Bo Markussen


# import data from dobson package
data(beetle)
m0 <- glm(cbind(y,n-y)~x, data=beetle, family=binomial(link="cloglog"))
# ED50 computation
summary(emmeans_ED(m0, tran="log10"), type="response")
# Visualization using the tidyverse
library(tidyverse)
LCL <- Vectorize(function(y,n) binom.test(y,n)$conf.int[1])
UCL <- Vectorize(function(y,n) binom.test(y,n)$conf.int[2])
beetle <- mutate(beetle, LCL=LCL(y,n), UCL=UCL(y,n))
emmeans_ED(m0, p=seq(0.001, 0.999, length.out=100), tran="log10") %>%
  summary(type="response") %>% as.data.frame() %>%
mutate(probability=as.numeric(as.character(probability))) %>%
ggplot(aes(x=probability, y=response, ymin=asymp.LCL, ymax=asymp.UCL)) +
  geom_ribbon(alpha=0.2, fill="blue") + geom_line() +
  xlab("Death probability") +
  ylab("Expected dose CS[2]-mg/l") +
  geom_errorbarh(aes(xmin=LCL, xmax=UCL,y=10^x), beetle, inherit.aes=FALSE) +
  geom_point(aes(x=y/n, y=10^x), beetle, inherit.aes=FALSE)
**minimum**

---

**Minimum between factors**

**Description**

minimum finds the minimum of two factors, i.e. the finest factors that is coarser than both of the factors.

**Usage**

minimum(x, y, concatenate.names = TRUE)

**Arguments**

x  
vector that will be interpreted as a factor.

y  
vector that will be interpreted as a factor.

concatenate.names  
boolean. If TRUE then the levels of the minimum are constructed as the concatenation of the levels for x and y. If FALSE then the levels of the minimum are given as numbers. Defaults to TRUE.

**Value**

A factor with the minimum.

**Author(s)**

Bo Markussen

**Examples**

```r
x <- rep(c("boy","girl","adult","adult"),4)
y <- rep(c("child","child","man","woman"),4)
minimum(x,y)
minimum(x,y,FALSE)
```
power.chisq.test.simulate

Simulate power of Chi-squared tests with conditioning

Description

power.chisq.test.simulate simulates power for tests for 2-way contingency tables based on the Pearson Chi-squared test statistics by simulation under 4 different conditioning scenarios.

Usage

```r
power.chisq.test.simulate(
  x,
  conditioning = "total",
  x0 = NULL,
  sig.level = 0.05,
  B = 10000
)
```

Arguments

- **x**: matrix specifying the alternative distribution of the contingency table.
- **conditioning**: character string specifying the simulation scenario. Defaults to "total". Other possible scenarios are "row", "col", and "both".
- **x0**: matrix specifying the null distribution. Defaults to NULL, in which case the null is estimated from the alternative x.
- **sig.level**: significance level used in test. Defaults to 0.05.
- **B**: integer specifying the number of replicates used in the Monte Carlo test. Defaults to 10000.

Details

Using conditioning="both" corresponds to selecting simulate.p.value=TRUE in chisq.test. However, conditioning on both row and column marginals appears to be rarely justified in real data. Instead conditioning="total" is the correct choice for testing independence. Similarly, conditioning="row" is recommended when the row marginals e.g. are fixed by experimental design. Both the alternative and the null are simulated under the parametric scenario estimated from the data matrix x. This possibly induces a discrepancy with chisq.test.simulate, where the null also is simulated from the specific data instance. Thus, the problem is that the null distribution depends on the model parameters.

Value

An object of class "power.htest".
Note
The code has not been optimized for speed, and might be slow.

Author(s)
Bo Markussen

See Also
chisq.test.simulate

Examples
# The Avadex dataset
Xobs <- matrix(c(2,3,6,40),2,2)ownames(Xobs) <- c("Avadex +","Avadex -")
colnames(Xobs) <- c("Tumor +","Tumor -")

# In this example only the rows appear to be fixed by experimental design.
power.chisq.test.simulate(Xobs,"row")
power.chisq.test.simulate(Xobs,"total")
Index

* design
  DD, 3

* htest
  chisq.test.simulate, 2
  power.chisq.test.simulate, 10

* manip
  DD, 3
  minimum, 9

arrow, 6

chisq.test, 2, 10
chisq.test.simulate, 2, 10, 11

DD, 3, 4, 7
designDiagram-class, 4

emmmeans, 7, 8
emmeans_ED, 7

formula, 7

glm, 7

minimum, 4, 9

plot.designDiagram, 4
plot.designDiagram
  (designDiagram-class), 4
power.chisq.test.simulate, 10
print.designDiagram
  (designDiagram-class), 4

summary.designDiagram
  (designDiagram-class), 4

update.designDiagram
  (designDiagram-class), 4