Package ‘Medbetareg’

August 23, 2017

Type Package

Title Generalized Beta Regression to Elicit Conditional Distributions of Medical Variables

Version 1.0

Date 2017-08-23

Author Alessandro Magrini

Maintainer Alessandro Magrini <alessandro.magrini@for.unipi.it>

Description Software for the method proposed in the article 'Generalized Beta Regression to Elicit Conditional Distributions of Medical Variables' by A. Magrini, D. Luciani and F. M. Stefanini, to be appeared on Austrian Journal of Statistics.

Depends BB, betareg

License GPL-2

R topics documented:

`Medbetareg-package` *Generalized Beta Regression to Elicit Conditional Distributions of Medical Variables*

Description

Software for the method proposed in the article ’Generalized Beta Regression to Elicit Conditional Distributions of Medical Variables’ by A. Magrini, D. Luciani and F. M. Stefanini, to be appeared on Austrian Journal of Statistics.

Details

Package:	Medbetareg
Type:	Package
Version:	1.0
Date:	2017-08-23
License:	GPL-2

The elicitation of an informative prior distribution on parameters of a univariate conditional model is typically diffucult when the response is a medical continuous variable, because some level of statistical training is required to a medical expert for interpreting parameters and for retrieving appropriate quantitative information about them. Magrini et al. (2017) proposed a Generalized Beta regression where parameter elicitation is performed by establishing a correspondence among measured values expressed as relative positions within intervals with a clinical interpretation, regardless the original scales of variables.

Author(s)

Alessandro Magrini <alessandro.magrini@for.unipi.it>

References

A. Magrini, D. Luciani, and F. M. Stefanini (2017). "Generalized Beta Regression to Elicit Conditional Distributions of Medical Variables". To be appeared on *Austrian Journal of Statistics*.

`eValFun.plot` *Plot marginal conditional expected value functions*

Description

Plot the conditional expected value as a function of a continuous explanatory variable given that all other explanatory variables take their respective neutral values.

Usage

```
eValFun.plot(B=1, yscale=NULL, xscale=NULL)
```

Arguments

`B`	The regression coefficient associated to the explanatory variable. Default is 1.
`yscale`	A vector including the 4 cut points of the response. If omitted, rescaled cut points are used.
`xscale`	A vector including the 4 cut points of the explanatory variable. If omitted, rescaled cut points are used.

Value

An object of class `plot`.

Examples

```
eValFun.plot(B=1)
eValFun.plot(B=1, yscale=c(0,15,25,40), xscale=c(0,0,30,100))
```

---

invRescale *Inverse rescaling*

Description

Inverse rescaling for a set of values of a medical variable.

Arguments

v	A numerical value or a numerical vector containing the values to be rescaled.
vscale	A vector of 4 cutpoints if the variable is continuous, or NA if the variable is binary.

Value

A numerical value or a numerical vector.

Examples

```
invRescale(v=seq(-1.5,1.5,length=21), vscale=c(0,15,25,40))
```

---

newPrior *Elicitation of the joint prior distribution*

Description

Computation of the joint prior distribution on parameters from expert assessments. Note: the original scales of variables are not required.

Usage

```
newPrior(assess.code, nrep = 100, digits = 6)
```

Arguments

assess.code	The code for expert assessments. See details.
nrep	The number of bootstrap replications to approximate the prior covariance matrix. Default is 100.
digits	The number of decimals to be shown. Default is 6.

Details

The code for expert assessments must be a character string containing several instructions. Each instruction is delimited from the other by a semicolumn or a line break, and must be of one of the types detailed below.

- Features of the response variable. Syntax:

```
RESP nameOfResponse scaleOfResponse
```

where `RESP` is the keyword of the instruction, `nameOfResponse` is the name of the response variable, `scaleOfResponse` is the set of its cutpoints indicated within round brackets and separated by commas. There must be one and only one of this instruction;

- Features of a continuous explanatory variable. Syntax:

```
CEV nameOfEV scaleOfEV SelectedProp SelectedRange expectedProp expectedRange nCase
```

where `CEV` is the keyword of the instruction, `nameOfEV` is the name of the continuous explanatory variable, `scaleOfEV` is the set of its cutpoints indicated within round brackets and separated by commas, `SelectedProp` and `SelectedRange` are the relative position and the range selected by the expert, `expectedProp` and `expectedRange` are the relative position and the range of the expected value of the response assessed by the expert, `nCase` is the number of patient cases on which the assessment is based. There must be any instances of this instruction, even zero;

- Features of a binary explanatory variable. Syntax:

```
BEV nameOfEV expectedProp expectedRange nCase
```

where `BEV` is the keyword of the instruction, `nameOfEV` is the name of the binary explanatory variable, `expectedProp` and `expectedRange` are the relative position and the range of the expected value of the response assessed by the expert, `nCase` is the number of patient cases on which the assessment is based. There must be any instances of this instruction, even zero;

- Features of an interaction among a set of explanatory variables. Syntax:

```
INTER nameOfEV_first ... nameOfEV_last expectedProp expectedRange nCase
```

where `INTER` is the keyword of the instruction, `nameOfEV_first ... nameOfEV_last` are the names of the interacting explanatory variables separated by spaces, `expectedProp` and `expectedRange` are the relative position and the range of the expected value of the response assessed by the expert, `nCase` is the number of patient cases on which the assessment is based. There must be any number of instances of this instruction, even zero; There must be any instances of this instruction, even zero;

- Assessments to determine the precision parameter. Syntax:

```
TAU expectedProp expectedRange
```

where `TAU` is the keyword of the instruction, `expectedProp` and `expectedRange` are the relative position and the range of the expected value of the response when all explanatory variables take their respectove neutral values. There must be one and only one of this instruction.

Value

An object of class `mbr`, that is a list with the following components:

- `scales`: a named list containing, for each variable in the model, either a vector of 4 cutpoints if the variable is continuous, or `NA` if the variable is binary. The first component of the list is referred to the response variable.
- `scales`: a list containing the cutpoints for each variable.
- `mean`: a vector containing the prior mean of parameters.
- `vcov`: a matrix containing the prior covariance matrix of parameters.

S3 methods available for the mbr class:

- print;
- summary.

See Also

predictive

Examples

```
assess.test <- 'RESP RespRate (0,15,25,40);
  CEV intraShunt (0,2,5,100) 0.5 hp 0.5 hp 5;
  CEV deadSpace (0,0,30,100) 0.5 hp 0.5 hp 5;
  CEV extraShunt (0,0,5,100) 0.5 hp 0.5 hp 5;
  CEV redAlvSpace (0,0,5,100) 0.5 hp 0.5 hp 5;
  BEV Panic 0.25 hp 25;
  BEV Neuromusc 0.6 lp 100;
  INTER intraShunt deadSpace 0.9 hp 5;
  TAU 0.3 n'
###  NOT RUN: replicate the results in Magrini et al. (2017)
# set.seed(10)
# prior.test <- newPrior(assess.test, nrep=5000)
#############
prior.test <- newPrior(assess.test, nrep=100)
```

predictive — *Plot predictive distributions*

Description

Plot a predictive distribution chosen by the user.

Usage

```
predictive(x, cnfg, nrep = 500, title = NULL)
```

Arguments

x	An object of class mbr.
cnfg	A numerical vector representing a configuration of explanatory variables (original scale).
nrep	An integer positive number representing the number of Monte Carlo replications. Default is 500.
title	The title of the plot. If NULL (the default), the configuration of explanatory variables is shown as a title.

Value

An object of class mbr.

See Also

newPrior

Examples

```
assess.test <- 'RESP RespRate (0,15,25,40);
  CEV intraShunt (0,2,5,100) 0.5 hp 0.5 hp 5;
  CEV deadSpace (0,0,30,100) 0.5 hp 0.5 hp 5;
  CEV extraShunt (0,0,5,100) 0.5 hp 0.5 hp 5;
  CEV redAlvSpace (0,0,5,100) 0.5 hp 0.5 hp 5;
  BEV Panic 0.25 hp 25;
  BEV Neuromusc 0.6 lp 100;
  INTER intraShunt deadSpace 0.9 hp 5;
  TAU 0.3 n'
###  NOT RUN: replicate the results in Magrini et al. (2017)
# set.seed(10)
# prior.test <- newPrior(assess.test, nrep=5000)
# set.seed(10)
# predictive(prior.test, xcfg1, nrep=50000, title="Configuration 1")
# set.seed(10)
# predictive(prior.test, xcfg2, nrep=50000, title="Configuration 2")
# set.seed(10)
# predictive(prior.test, xcfg3, nrep=50000, title="Configuration 3")
# set.seed(10)
# predictive(prior.test, xcfg4, nrep=50000, title="Configuration 4")
#############
prior.test <- newPrior(assess.test, nrep=100)
predictive(prior.test, c(5,30,5,5,0,0), nrep=500)
```

rescale *Rescaling procedure*

Description

Rescaling for a set values of a medical variable.

Usage

```
rescale(v, vscale)
```

Arguments

v	A numerical value or a numerical vector containing the values to be rescaled.
vscale	A vector of 4 cutpoints if the variable is continuous, or NA if the variable is binary.

Value

A numerical value or a numerical vector.

Examples

```
rescale(v=seq(0,40,length=21), vscale=c(0,15,25,40))
```

Index