

Package ‘lsm’

August 30, 2018

Type Package

Title Estimation of the log Likelihood of the Saturated Model

Version 0.1.8

Date 2018-08-30

Author Humberto Llinas [aut],
Omar Fabregas [aut],
Jorge Villalba [aut, cre]

Maintainer Jorge Villalba <jlvial1191@gmail.com>

Description When the values of the outcome variable Y are either 0 or 1, the function `lsm()` calculates the estimation of the log likelihood in the saturated model. This model is characterized by Llinas (2006, ISSN:2389-8976) in section 2.3 through the assumptions 1 and 2. The function `LogLik()` works (almost perfectly) when the number of independent variables K is high, but for small K it calculates wrong values in some cases. For this reason, when Y is dichotomous and the data are grouped in J populations, it is recommended to use the function `lsm()` because it works very well for all K .

Depends R (>= 3.1.0),stats

Encoding UTF-8

URL <https://github.com/jlvial1191/lsm>

License MIT + file LICENSE

LazyData TRUE

RoxygenNote 6.1.0

Collate 'lsm.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2018-08-30 08:12:13 UTC

R topics documented:

lsm	2
Index	5

Description

When the values of the outcome variable Y are either 0 or 1, the function `lsm()` calculates the estimation of the log likelihood in the saturated model. This model is characterized by Llinas (2006, ISSN:2389-8976) in section 2.3 through the assumptions 1 and 2. If Y is dichotomous and the data are grouped in J populations, it is recommended to use the function `lsm()` because it works very well for all K .

Usage

```
lsm(formula, data)
```

Arguments

<code>formula</code>	An expression of the form $y \sim \text{model}$, where y is the outcome variable (binary or dichotomous: its values are 0 or 1).
<code>data</code>	an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in <code>data</code> , the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>lsm()</code> is called.

Details

The saturated model is characterized by the assumptions 1 and 2 presented in section 2.3 by Llinas (2006, ISSN:2389-8976).

Value

`lsm` returns an object of class "lsm".

An object of class "lsm" is a list containing at least the following components:

<code>log_Likelihood</code>	Estimation of the log likelihood.
<code>populations</code>	Total number J of populations in the model.
<code>z_j</code>	Value of Z_j (the sum of the observations in the j th population).
<code>n_j</code>	Number of the observations in the j th population.
<code>p_j</code>	Estimation of p_j in the j th population.
<code>fitted.values</code>	Value of the <code>log_Likelihood</code> in the j th population.
<code>v_j</code>	Variance of the Bernoulli variables in the j th population.
<code>m_j</code>	Expected value of Z_j .
<code>V_j</code>	Variance of Z_j .
<code>V</code>	Variance and covariance matrix of Z , the vector that contains all the Z_j .

S_p Score vector of the model.
 I_p Information matrix of the model.
 Zast_j Standardized variable of Zj.

Author(s)

Humberto Llinas Solano [aut], Universidad del Norte, Barranquilla-Colombia \ Omar Fabregas Cera [aut], Universidad del Norte, Barranquilla-Colombia \ Jorge Villalba Acevedo [cre, aut], Cartagena-Colombia.

References

- [1] Humberto Jesus Llinas. (2006). Accuracies in the theory of the logistic models. *Revista Colombiana De Estadística*, 29(2), 242-244.
 [2] Hosmer, D. (2013). *Wiley Series in Probability and Statistics Ser. : Applied Logistic Regression* (3). New York: John Wiley & Sons, Incorporated.

Examples

```
# Hosmer, D. (2013) page 3: Age and coranary Heart Disease (CHD) Status of 20 subjects:

AGE <- c(20, 23, 24, 25, 25, 26, 26, 28, 28, 29, 30, 30, 30, 30, 30, 30, 30, 32, 33, 33)
CHD <- c(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0)

data <- data.frame (CHD, AGE)
lsm(CHD ~ AGE , data)

# Other case.

y <- c(0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1)
x1 <- c(2, 2, 2, 2, 2, 5, 5, 5, 5, 6, 6, 6, 8, 8, 11, 11, 11, 1)
x2 <- c(3, 3, 3, 3, 3, 6, 6, 6, 6, 8, 8, 8, 9, 9, 12, 12, 12, 12)
x3 <- c(4, 4, 4, 4, 4, 7, 7, 7, 7, 9, 9, 9, 10, 10, 13, 13, 13, 13)
x4 <- c(1, 1, 1, 1, 1, 9, 9, 9, 9, 10, 10, 10, 4, 4, 2, 2, 2, 2)
x5 <- c(32, 32, 32, 32, 32, 20, 20, 20, 20, 21, 21, 21, 19, 19, 16, 16, 16, 16)
x6 <- c(15, 15, 15, 15, 15, 18, 18, 18, 18, 16, 16, 16, 25, 25, 20, 20, 20, 20)
x7 <- c(28, 28, 28, 28, 28, 23, 23, 23, 23, 32, 32, 32, 24, 24, 32, 32, 32, 32)
x8 <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0)
x9 <- c(6, 6, 6, 6, 6, 10, 10, 10, 10, 11, 11, 11, 7, 7, 21, 21, 21, 21)
x10 <- c(5, 5, 5, 5, 5, 6, 6, 6, 6, 7, 7, 7, 7, 7, 8, 8, 8, 8)

data <- data.frame (y, x1, x2, x3, x4, x5, x6, x7, x8, x9, x10)
lsm(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10, data)

## For more ease, use the following notation.
lsm(y~., data)

## Other case.

y <- as.factor(c(1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1))
x1 <- as.factor(c(2, 2, 2, 5, 5, 5, 5, 8, 8, 11, 11, 11))
```

```
x2 <- as.factor(c(3, 3, 3, 6, 6, 6, 6, 9, 9, 12, 12, 12))
x3 <- as.factor(c(4, 4, 4, 7, 7, 7, 7, 10, 10, 13, 13, 13))
x4 <- as.factor(c(1, 1, 1, 9, 9, 9, 9, 4, 4, 2, 2, 2))
x5 <- as.factor(c(5, 5, 5, 6, 6, 6, 6, 7, 7, 8, 8, 8))

data <- data.frame (y, x1, x2, x3, x4, x5)
lsm(y ~ x1 + x2 + x3 + x4 + x5, data)

## For more ease, use the following notation.
lsm(y~., data)
```

Index

lsm, [2](#)