Package: generalhoslem (via r-universe)

August 31, 2024

Type Package

Title Goodness of Fit Tests for Logistic Regression Models

Version 1.3.4

Date 2019-06-01

Author Matthew Jay [aut, cre]

Maintainer Matthew Jay <matthew.jay.15@ucl.ac.uk>

Description Functions to assess the goodness of fit of binary,
multinomial and ordinal logistic models. Included are the
Hosmer-Lemeshow tests (binary, multinomial and ordinal) and the
Lipsitz and Pulkstenis-Robinson tests (ordinal).

Depends reshape, MASS, R (>= 3.3.1)

License GPL-2

Suggests nnet, mlogit, ordinal

Repository https://matthewjay15.r-universe.dev

RemoteUrl https://github.com/matthewjay15/generalhoslem

RemoteRef HEAD

RemoteSha 96fab3468e9f2579c9a02fb4c836e8ca99011f38

Contents

generalhoslem-package *Goodness of Fit Tests for Logistic Regression Models*

Description

Functions to assess the goodness of fit of binary, multinomial and ordinal logistic models. Included are the Hosmer-Lemeshow tests (binary, multinomial and ordinal) and the Lipsitz and Pulkstenis-Robinson tests (ordinal).

Details

This package was not yet installed at build time.

Index: This package was not yet installed at build time.

Author(s)

Matthew Jay [aut, cre] Maintainer: Matthew Jay <matthew.jay.15@ucl.ac.uk>

References

- Fagerland MW, Hosmer DW, Bofin AM. Multinomial goodness-of-fit tests for logistic regression models. Statistics in Medicine 2008;27(21):4238-53.

- Fagerland MW, Hosmer DW. A goodness-of-fit test for the proportional odds regression model. Statistics in Medicine 2013;32:2235-2249.

- Fagerland MW, Hosmer DW. Tests for goodness of fit in ordinal logistic regression models. Journal of Statistical Computation and Simulation 2016. DOI: 10.1080/00949655.2016.1156682.

- Fagerland MW, Hosmer DW. How to test for goodness of fit in ordinal logistic regression models. The Stata Journal 2017;17(3):668-686.

- Hosmer DW, Lemeshow S, Sturdivant RX. Applied Logistic Regression, 3rd Edition. 2013. New York, USA: John Wiley and Sons.

- Lipsitz SR, Fitzmaurice GM, Molenberghs G. Goodness-of-Fit Tests for Ordinal Response Regression Models. Journal of the Royal Statistical Society 1996;45(2):175-190.

- Pulkstenis E, Robinson TJ. Goodness-of-fit tests for ordinal response regression models. Statistics in Medicine 2004;23:999-1014.

---

lipsitz.test	*Lipsitz goodness of fit test for ordinal logistic models.*

Description

Performs the Lipsitz goodness of fit test for ordinal logistic regression models.

Usage

```
lipsitz.test(model, g = 10)
```

Arguments

model	an ordinal logistic regression model. Must be an object of class polr or clm.
g	number of quantiles of risk, defaults to 10.

Details

The Lipsitz test is a goodness of fit test for ordinal response logistic regression models. It involves binning the observed data into equally sized g groups based on an ordinal response score. This score is computed by summing the predicted probabilities of each subject for each outcome level multiplied by equally spaced integer weights. The user can specify the number of groups by assigning an integer value to g, which is 10 by default.

Given this partitioning of the data, dummy variables, I, are derived such that, for each group, I = 1 if the subject is in region g and I = 0 if not. The model is then re-fit with these dummy variables. If the model has good fit, then the coefficients for all these dummy variables simultaneously = 0. Lipsitz et al (1996) suggest that likelihood ratio, Wald or score tests can be used; lipsitz.test just uses the likelihood ratio test with g-1 degrees of freedom.

Note that the outcome variable MUST be converted to a factor before running the model. Using as.factor() within the model function will cause an error because of the way in which lipsitz.test uses the update() function to re-fit the model.

It is recommended (Fagerland and Hosmer, 2016) that the Lipsitz test be run alongside the ordinal Hosmer-Lemeshow test (logitgof) and the Pulkstenis-Robinson tests (pulkrob.chisq and pulkrob.deviance).

Value

A list of class htest containing:

statistic	the value of the likelihood ratio statistic.
parameter	degrees of freedom used.
p.value	the p-value.
method	a character string indicating the name of the test.
data.name	a character string indicating the model formula used.
newmoddata	a data frame containing the data used in fitting the updated model (the original data plus the dummy variables).
predictedprobs	a data frame of predicted probabilities from the original model.

Author(s)

Matthew Alexander Jay

References

- Fagerland MW, Hosmer DW. A goodness-of-fit test for the proportional odds regression model. Statistics in Medicine 2013;32:2235-2249.
- Fagerland MW, Hosmer DW. Tests for goodness of fit in ordinal logistic regression models. Journal of Statistical Computation and Simulation 2016. DOI: 10.1080/00949655.2016.1156682.
- Lipsitz SR, Fitzmaurice GM, Molenberghs G. Goodness-of-Fit Tests for Ordinal Response Regression Models. Journal of the Royal Statistical Society 1996;45(2):175-190.

See Also

logitgof, pulkrob.chisq.

Examples

```
data(mtcars)
mtcars$gear <- as.factor(mtcars$gear) # Outcome must be converted to factor
                                      # before fitting model otherwise
                                      # lipsitz.test() will fail.
mod1 <- polr(gear ~ mpg + cyl, data = mtcars)
lipsitz.test(mod1)

library(ordinal)
mod2 <- clm(gear ~ mpg, data = mtcars)
lipsitz.test(mod2)
```

---

logitgof   *Hosmer-Lemeshow Tests for Logistic Regression Models*

Description

Performs the Hosmer-Lemeshow goodness of fit tests for binary, multinomial and ordinal logistic regression models.

Usage

```
logitgof(obs, exp, g = 10, ord = FALSE)
```

Arguments

obs	a vector of observed values. See details.
exp	expected values fitted by the model. See details.
g	number of quantiles of risk, 10 by default.
ord	logical indicating whether to run the ordinal version, FALSE by default.

Details

The Hosmer-Lemeshow tests The Hosmer-Lemeshow tests are goodness of fit tests for binary, multinomial and ordinal logistic regression models. `logitgof` is capable of performing all three. Essentially, they compare observed with expected frequencies of the outcome and compute a test statistic which is distributed according to the chi-squared distribution. The degrees of freedom depend upon the number of quantiles used and the number of outcome categories. A non-significant p value indicates that there is no evidence that the observed and expected frequencies differ (i.e., evidence of good fit).

Binary version If obs is a vector of 1s and 0s or a factor vector with 2 levels, then the binary version of the test is run. `exp` must be the fitted values obtained from the model, which can be accessed using the `fitted()` function.

Multinomial version If obs is a factor with three or more levels and `ord = FALSE`, the multinomial version of the test is run. If using the `mlogit` package to run a model, ensure `outcome = FALSE` in the `fitted()` function. See examples.

Ordinal version If obs is a factor with three or more levels and `ord = TRUE`, the ordinal version of the test is run. See examples for how to extract fitted values from models constructed using `MASS::polr` or `oridinal::clm`.

Note that Fagerland and Hosmer (2013) point out that the model needs to have at least as many covariate patterns as groups. This is achieved easily where there are continuous predictors or several categorical variables. This test will not be valid where there is only one or two categorical predictor variables. Fagerland and Hosmer (2016) also recommend running the Hosmer-Lemeshow test for ordinal models alongisde the Lipsitz test (`lipsitz.test`) and Pulkstenis-Robinson tests (`pulkrob.chisq` and `pulkrob.deviance`), as each detects different types of lack of fit.

Finally, it has been observed that the results from this implementation of the ordinal Hosmer-Lemeshow test and the Lipsitz test are slightly different from the Fagerland and Hosmer (2017) Stata implementation. It is not not yet clear why this is but is under investigation. The discrepancy is ver minor.

Value

A list of class `htest` containing:

`statistic`	the value of the relevant test statistic.
`parameter`	the number of degrees of freedom used.
`p.value`	the p-value.
`method`	a character string indicating whether the binary or multinomial version of the test was performed.
`data.name`	a character string containing the names of the data passed to obs and exp.
`observed`	a table of observed frequencies with g rows. Either an `xtabs` generated table (used in the binary version) or a `cast` generated data frame (multinomial version).
`expected`	a table of expected frequencies with g rows. Either an `xtabs` generated table or a `cast` generated data frame.
`stddiffs`	a table of the standardised differences. See Hosmer, Lemeshow and Sturdivant (2013), p 162.

Author(s)

Matthew Alexander Jay, with code adapted from `ResourceSelection::hoslem.test` by Peter Solymos.

References

- Fagerland MW, Hosmer DW, Bofin AM. Multinomial goodness-of-fit tests for logistic regression models. Statistics in Medicine 2008;27(21):4238-53.

- Fagerland MW, Hosmer DW. A goodness-of-fit test for the proportional odds regression model. Statistics in Medicine 2013;32:2235-2249.

- Fagerland MW, Hosmer DW. Tests for goodness of fit in ordinal logistic regression models. Journal of Statistical Computation and Simulation 2016. DOI: 10.1080/00949655.2016.1156682.

- Fagerland MW, Hosmer DW. How to test for goodness of fit in ordinal logistic regression models. The Stata Journal 2017;17(3):668-686.

- Hosmer DW, Lemeshow S, Sturdivant RX. Applied Logistic Regression, 3rd Edition. 2013. New York, USA: John Wiley and Sons.

See Also

[`lipsitz.test`](), [`pulkrob.chisq`]().

Examples

```
### The mtcars dataset is a terrible example to use due to its small
### size - some of the models will return warnings as a result.
## Binary model
# 1/0 coding
data(mtcars)
mod1 <- glm(vs ~ cyl + mpg, data = mtcars, family = binomial)
logitgof(mtcars$vs, fitted(mod1))

# factor name coding
mtcars$engine <- factor(ifelse(mtcars$vs==0, "V", "S"), levels = c("V", "S"))
mod2 <- glm(engine ~ cyl + mpg, data = mtcars, family = binomial)
logitgof(mtcars$engine, fitted(mod2))

## Multinomial model
# with nnet
library(nnet)
mod3 <- multinom(gear ~ mpg + cyl, data = mtcars)
logitgof(mtcars$gear, fitted(mod3))

# with mlogit
library(mlogit)
data("Fishing", package = "mlogit")
Fish <- mlogit.data(Fishing, varying = c(2:9), shape = "wide", choice = "mode")
mod4 <- mlogit(mode ~ 0 | income, data = Fish)
logitgof(Fishing$mode, fitted(mod4, outcome = FALSE))

## Ordinal model
```

```
# polr in package MASS
mod5 <- polr(as.factor(gear) ~ mpg + cyl, data = mtcars)
logitgof(mtcars$gear, fitted(mod5), g = 5, ord = TRUE)

# clm in package ordinal
library(ordinal)
mtcars$gear <- as.factor(mtcars$gear)
mod6 <- clm(gear ~ mpg + cyl, data = mtcars)
predprob <- data.frame(mpg = mtcars$mpg, cyl = mtcars$cyl)
fv <- predict(mod6, newdata = predprob, type = "prob")$fit
logitgof(mtcars$gear, fv, g = 5, ord = TRUE)
```

---

pulkrob	*Pulkstenis-Robinson goodness of fit tests for ordinal response models.*

Description

Performs the Pulkstenis-Robinson goodness of fit chi-squared and deviance tests for ordinal logistic regression models.

Usage

```
pulkrob.chisq(model, catvars)
pulkrob.deviance(model, catvars)
```

Arguments

model   an ordinal response model. Must be an object of class polr or clm.

catvars   a character vector containing the names of the categorical covariates.

Details

The Pulkstenis-Robinson tests are goodness of fit tests for ordinal logistic regression models. They are capable of accommodating models with continuous as well as categorical predictors. The data are partitioned according to observed covariate patterns using the categorical covariates only. Any unobserved covariate patterns are discarded. Only categorical predictors are used to avoid partitioning among an unacceptably high number of covariate patterns. Each subject is assigned an ordinal response score by summing the predicted probabilities of each subject for each outcome level multiplied by equally spaced integer weights. The covariate patterns are then split into two at the median score within each.

Based on this partitioning, observed and expected frequencies are calculated and the test statistic computed. This statistic is distributed by the chi-squared distribution with $(2I - 1)(J - 1) - k - 1$ degress of freedom, where I is the number of covariate patterns, J is the number of of response categories and k is the number of categorical variables in the model.

It is recommended (Fagerland and Hosmer, 2016) that the Pulkstenis-Robinson tests be run alongside the Hosmer-Lemeshow test (logitgof) and the Lipsitz test (lipsitz.test).

Value

A list of class htest containing:

statistic	the chi-squared or deviance statistic.
parameter	degrees of freedom used.
p.value	the p-value.
method	a character string indicating the name of the test.
data.name	a character string indicating the model formula used.
observed	a cast generated data frame of observed frequencies.
expected	a cast generated data frame of expected frequencies.
stddiffs	a table of the standardised differences. See Hosmer, Lemeshow and Sturdivant (2013), p 162.

Author(s)

Matthew Alexander Jay, with code from epiR::epi.cp() by Mark Stevenson.

References

- Fagerland MW, Hosmer DW. Tests for goodness of fit in ordinal logistic regression models. Journal of Statistical Computation and Simulation 2016. DOI: 10.1080/00949655.2016.1156682.
- Hosmer DW, Lemeshow S, Sturdivant RX. Applied Logistic Regression, 3rd Edition. 2013. New York, USA: John Wiley and Sons.
- Pulkstenis E, Robinson TJ. Goodness-of-fit tests for ordinal response regression models. Statistics in Medicine 2004;23:999-1014.

See Also

logitgof, lipsitz.test.

Examples

```
data(mtcars)

# using polr
mod1 <- polr(as.factor(gear) ~ mpg + cyl + vs, data = mtcars)
pulkrob.chisq(mod1, c("vs"))
pulkrob.deviance(mod1, c("vs"))

# using clm - ensure outcome variable is a factor before fitting the model
library(ordinal)
mtcars$gear <- as.factor(mtcars$gear)
mod2 <- clm(gear ~ mpg + cyl + vs, data = mtcars)
pulkrob.chisq(mod2, c("vs"))
pulkrob.deviance(mod2, c("vs"))
```

Index