Package ‘clarkeTest’

May 16, 2024

Type Package

Title Distribution-Free Tests of Non-Nested Models

Version 0.2.0

Description Implementation of Clarke's distribution-free test of non-nested models. Currently supported model functions are: lm(), glm() ('binomial', 'poisson', 'negative binomial' links), polr() ('MASS'), clm() ('ordinal'), and multinom() ('nnet'). For more information on the test, see Clarke (2007) <doi:10.1093/pan/mpm004>.

Depends R (>= 3.5.0)

Suggests MASS, ordinal, nnet

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

NeedsCompilation no

Author Dave Armstrong [aut, cre],
Brenton Kenkel [aut]

Maintainer Dave Armstrong <davearmstrong.ps@gmail.com>

Repository CRAN

Date/Publication 2024-05-16 14:40:02 UTC

R topics documented:

`clarke_test` *Clarke Test*

Description

‘clarke_test‘ returns results from Kevin Clarke’s distribution-free test of non-nested models.

Usage

```
clarke_test(model1, model2, level=0.05, digits=2)
```

Arguments

`model1`	A fitted statistical model of a supported class
`model2`	A fitted statistical model of a supported class whose dependent variable is the same as that of `model1`
`level`	Numeric: significance level for the test.
`digits`	Integer: number of digits to print

Details

‘clarke_test‘ is a more modularized version of the [clarke()] function from the [games] package. The main innovation is that the ‘nonnest‘ function calls a generic ‘indivLogLiks‘ function, so additional methods can be easily written for different classes of models. The function currently supports binomial, poisson and negative binomial GLMs, ordinal models estimated with either `polr` from the `MASS` package or `clm` from the `ordinal` package and multinomial models estimated with either `multinom` from the `nnet` package. Users can also write new methods for both `indivLogLiks` and `nparams` that would get called by the generic function.

Value

Typical use will be to run the function interactively and examine the printed output. The functions return an object of class `nonnest.test`, which is a list containing:

`stat` The test statistic

`level` Significance level for the test

`digits` Number of digits to print

`loglik1` Vector of observationwise log-likelihoods for `model1`

`loglik2` Vector of observationwise log-likelihoods for `model2`

`nparams` Integer vector containing the number of parameters fitted in `model1` and `model2` respectively

`nobs` Number of observations of the dependent variable being modeled

An object of class `nonnest.test` with the following values:

stat The number of times model 1 is better than model 2

test Will always be "clarke".

level The chosen confidence level for the test

digits The number of digits to print

loglik1 Individual log-likelihoods for model 1

loglik2 Individual log-likelihoods for model 2

nparams A vector giving the number of parameters in models 1 and 2, respectively

nobs Number of observations in the model

Author(s)

Brenton Kenkel (<brenton.kenkel@gmail.com>) modified by Dave Armstrong (<dave@quantoid.net>)

References

Kevin Clarke. 2007. "A Simple Distribution-Free Test for Nonnested Hypotheses." *Political Analysis* 15(3): 347–363.

Examples

```
data(conflictData)
## Linear Model
lm1 <- lm(riots ~ log(rgdpna_pc) + log(pop*1000) +
    polity2, data=conflictData)
lm2 <- lm(riots ~ rgdpna_pc + pop +
    polity2, data=conflictData)
clarke_test(lm1, lm2)

## Binomial GLM
glm1 <- glm(conflict_binary ~ log(rgdpna_pc) +
          log(pop*1000) + polity2, data=conflictData,
          family=binomial)
glm2 <- glm(conflict_binary ~ rgdpna_pc + pop +
          polity2, data=conflictData,
          family=binomial)
clarke_test(glm1, glm2)

## Poisson GLM
glm1a <- glm(riots ~ log(rgdpna_pc) +
              log(pop*1000) + polity2,
             data=conflictData,
             family=poisson)
glm2a <- glm(riots ~ rgdpna_pc + pop +
              polity2, data=conflictData,
            family=poisson)
clarke_test(glm1a, glm2a)

## Negative Binomial GLM
library(MASS)
glm1b <- glm.nb(riots ~ log(rgdpna_pc) +
               log(pop*1000) + polity2,
               data=conflictData)
glm2b <- glm.nb(riots ~ rgdpna_pc + pop +
               polity2, data=conflictData)
clarke_test(glm1b, glm2b)

## Ordered Logit: polr
library(MASS)
ol1 <- polr(as.factor(Amnesty) ~ log(rgdpna_pc) +
                  log(pop*1000) + polity2,
                data=conflictData)
ol2 <- polr(as.factor(Amnesty) ~ scale(rgdpna_pc) +
            scale(pop) + polity2,
            data=conflictData)
clarke_test(ol1, ol2)

## Ordered Logit: clm
library(ordinal)
ol1a <- clm(as.factor(Amnesty) ~ log(rgdpna_pc) +
              log(pop*1000) + polity2,
            data=conflictData)
ol2a <- clm(as.factor(Amnesty) ~ scale(rgdpna_pc) +
            scale(pop) + polity2,
            data=conflictData)
clarke_test(ol1a, ol2a)

## Multinomial Logit: multinom

library(nnet)
ml1 <- multinom(as.factor(Amnesty) ~ log(rgdpna_pc) +
              log(pop*1000) + polity2,
            data=conflictData)
ml2 <- multinom(as.factor(Amnesty) ~ scale(rgdpna_pc) +
              scale(pop) + polity2,
            data=conflictData)
clarke_test(ml1, ml2)


## Multinomial Logit: multinom
```

conflictData *Conflict Data*

Description

A country-year dataset containing information on conflict and other country attributes. These data come from multiple sources and are simply for the purposes of demonstrating how the functions in the package work. The data contain the following variables

Usage

```
conflictData
```

Format

A data frame with 4381 rows and 9 variables

GWNo Gleditsch and Ward country number

Year year

StateName Country name

conflict_binary Binary indicator of conflict

polity2 Polity IV indicator of regime type

Amnesty Amnesty International Political Terror Scale Rating

riots Number of riots in each country-yar

pop Population in country (in thousands)

rgdpna_pc PWT measure of GDP/capita

`indivLogLiks` *Calculate individual log-likelihood values*

Description

Calculate individual log-likelihood values

Usage

```
indivLogLiks(model)

## S3 method for class 'glm'
indivLogLiks(model)

## S3 method for class 'lm'
indivLogLiks(model)

## S3 method for class 'orlm'
indivLogLiks(model)

## S3 method for class 'polr'
indivLogLiks(model)

## S3 method for class 'clm'
indivLogLiks(model)

## S3 method for class 'multinom'
```

```
indivLogLiks(model)

## S3 method for class 'negbin'
indivLogLiks(model)
```

Arguments

`model`	A statistical model object.

Details

The `indivLogLiks` function calls the appropriate method for calculating individual log likelihood values for the model. The function currently supports binomial, poisson and negative binomial GLMs, ordinal models estimated with either `polr` from the `MASS` package or `clm` from the `ordinal` package and multinomial models estimated with either `multinom` from the `nnet` package. Users can also write new methods for both `indivLogLiks` and `nparams` that would get called by the generic function.

For the purposes of the `clarke_test` function, the `indivLogLiks` functions are not intended to be called directly by the user.

Value

A vector of individual log-likelihood values for the model.

`nparams` *Find number of parameters in model*

Description

Finds the number of parameters that were estimated in each model.

Usage

```
nparams(model)

## S3 method for class 'glm'
nparams(model)

## S3 method for class 'lm'
nparams(model)

## S3 method for class 'orlm'
nparams(model)

## S3 method for class 'polr'
nparams(model)
```

```
## S3 method for class 'clm'
nparams(model)

## S3 method for class 'multinom'
nparams(model)

## S3 method for class 'negbin'
nparams(model)
```

Arguments

model	A statistical model object.

Details

The function funds the number of parameters generally by counting the number of estimated parameters in the model's output.

For the purposes of the clarke_test function, the nparams functions are not intended to be called directly by the user.

Value

A scalar giving the number of parameters estimated in the model.

print.nonnest.test *Print non-nested test results*

Description

Prints results of the clarke_test function.

Usage

```
## S3 method for class 'nonnest.test'
print(x, digits = x$digits, ...)
```

Arguments

x	A result from the ‘nonnest‘ function
digits	Number of digits to print in the output
...	Other arguments passed down to print

Value

Printed output that summarises the results of the clarke_test function.

Index