Package ‘mpcmp’

March 4, 2019

Type Package
Title Mean-Parametrized Conway-Maxwell Poisson (COM-Poisson) Regression
Version 0.1.3
Date 2019-02-14
Depends R (>= 2.10)
License GPL (>= 2)
Imports graphics, grDevices, stats
URL https://github.com/thomas-fung/mpcmp
Encoding UTF-8
LazyData true
RoxygenNote 6.1.0
Suggests knitr, rmarkdown
NeedsCompilation no
Author Thomas Fung [aut, cre],
Aya Alwan [aut],
Justin Wishart [aut],
Alan Huang [aut]
Maintainer Thomas Fung <thomas.fung@mq.edu.au>
Repository CRAN
Date/Publication 2019-03-04 17:20:03 UTC

R topics documented:

  mpcmp-package ....................................................... 2
  AIC.cmp ............................................................ 3
  attendance ......................................................... 3
mpcmp-package

Mean-parametrized Conway-Maxwell-Poisson Regression

Description

Mean-parametrized Conway-Maxwell-Poisson Regression

References


Akaike’s Information Criterion

Description

A function calculating Akaike’s Information Criterion (AIC) based on the log-likelihood value extracted from loglik.cmp, according to the formula -2*log-likelihood + k*npar, where npar represents the number of parameters in the fitted model, and k=2 for the usual AIC or k=log(n) (n being the number of observations) for the so-called BIC (Bayesian Information Criterion).

Usage

## S3 method for class 'cmp'
AIC(object, ..., k = 2)

Arguments

object an object class ‘cmp’ object, obtained from a call to glm.cmp
... other arguments passed to or from other methods (currently unused).
k numeric: the penalty per parameter to be used; the default k = 2 is the classical AIC.

Details

When comparing models fitted by maximum likelihood to the same data, the smaller the AIC or BIC, the better the fit.

Value

A numeric value with the corresponding AIC (or BIC, or ..., depends on k).

See Also

logLik.cmp, nobs.cmp, glm.cmp

attendance Attendance data set

Description

This data set gives the number of days absent from high school and the gender, maths score (standardized score out of 100) and academic programme (‘General’, ‘Academic’ and ‘Vocational’) of 314 students sampled from two urban high schools. The attendance data frame has 314 observations on 5 variables.
Usage

data(attendance)

Format

A data frame with 314 observations on 5 variables.

id  Identifier
gender  gender
math  standardized math score out of 100
daysabs  number of days absent from high school
prog  academic programme (‘General’, ‘Academic’ and ‘Vocational’)

Source

http://www.ats.ucla.edu/stat/stata/dae/nb_data.dta

Examples

### For examples see example(glm.cmp)

<table>
<thead>
<tr>
<th>CBIND</th>
<th>Combine R Objects by Columns</th>
</tr>
</thead>
</table>

Description

Take a sequence of vector, matrix or data-frame arguments and combine them by columns. CBIND is used within the package over cbind to recycle the shorter arguments so that their number of rows would match.

Usage

CBIND(..., deparse.level = 1)

Arguments

... (generalized) vectors or matrices. These can be given as named arguments.

deparse.level  integer; deparse.level = 0 constructs no labels, deparse.level = 1 (the default) or > 1 constructs labels from the arguments names.
cmplrtest

Likelihood Ratio Test for nested COM-Poisson models

Description

Perform a likelihood ratio chi-squared test between nested COM-Poisson models. The test statistics is calculated as $2(\text{llik} - \text{llik}_0)$. The test statistics has degrees of freedom $r$ where $r$ is the difference in the number of parameters between the full and null models.

Usage

```r
cmplrtest(object1, object2, digits = 3)
```

Arguments

- `object1`: an object class `cmp`, obtained from a call to `glmNcmp`
- `object2`: an object class `cmp`, obtained from a call to `glmNcmp`
- `digits`: numeric; minimum number of significant digits to be used for most numbers.

Details

Obviously the comparison between two models will only be valid if they are fitted to the same data set.

References


See Also

`glmNcmp`, `updateNcmp`

Examples

```r
data(takeoverbids)

## Fit full model
M.bids.full <- glmNcmp(numbids ~ leglrest + rearest + finrest + whtknght + bidprem + insthold + size + sizesq + regulatn, data=takeoverbids)

## Fit null model; without whtknght
M.bids.null <- update(M.bids.full, .~.-whtknght)

## Likelihood ratio test for the nested models
cmplrtest(M.bids.full, M.bids.null) # order of objects is not important
```
Description

An accessor function used to extract model coefficients from a 'cmp' object. coefficients is an alias for coef.

Usage

```r
## S3 method for class 'cmp'
coef(object, ...)
```

Arguments

- `object`: an object class 'cmp' object, obtained from a call to glm cmp
- `...`: other arguments passed to or from other methods (currently unused).

Value

Coefficients extracted from the object object.

See Also

`fitted.cmp`, `residuals.cmp`, `glm.cmp`.

Description

Functions to approximate the various expected values for the COM-Poisson distribution via truncation. The standard COM-Poisson parametrization is being used here. The lambda and nu values are recycled to match the length of the longer one and that would determine the length of the results. Notice that the sum is hard coded to truncate at 100 so the approximation will be quite bad if the COM-Poisson has a large rate or mean.
comp_lambdas

Usage

comp_mean_logfactorialy(lambda, nu)
comp_mean_ylogfactorialy(lambda, nu)
comp_means(lambda, nu)
comp_variances(lambda, nu)
comp_variances_logfactorialy(lambda, nu)

Arguments

lambda, nu rate and dispersion parameters. Must be positives.

Value

comp_mean_logfactorialy gives the mean of \( \log(Y!) \).
comp_mean_ylogfactorialy gives the mean of \( y\log(Y!) \).
comp_means gives the mean of \( Y \).
comp_variances gives the variance of \( Y \).
comp_variances_logfactorialy gives the variance of \( \log(Y!) \).

Description

Given a particular mean parametrized COM-Poisson distribution i.e. \( \mu \) and \( \nu \), this function is used to find a lambda that can satisfy the mean constraint with a combination of bisection and Newton-Raphson updates. The function is also vectorized but will only update those that have not converged.

Usage

comp_lambdas(mu, nu, lambdalb = 1e-10, lambdaub = 1900,
maxlambdaiter = 1000, tol = 1e-06, lambdaint = 1)

Arguments

mu, nu mean and dispersion parameters. Must be straightly positive.
lambdalb, lambdaub numeric; the lower and upper end points for the interval to be searched for lambda(s).
maxlambdaiter numeric; the maximum number of iterations allowed to solve for lambda(s).

tol numeric; the convergence threshold. A lambda is said to satisfy the mean constraint if the absolute difference between the calculated mean and the corresponding mu values is less than tol.

lambdaint numeric vector; initial guess for lambda(s).

Value

The function returns the lambda value(s) that satisfies the mean constraint(s).

---

**comp_mu_loglik**

*Calculate the Log-Likelihood of the COM-Poisson model*

---

Description

A function to compute the log-likelihood of the COM-Poisson model.

Usage

```r
comp_mu_loglik(param, y, xx, offset)
```

Arguments

- **param** numeric vector: the model coefficients & the current value of nu. It is assumed that nu is in the last position of param.
- **y** numeric vector: response variable
- **xx** numeric matrix: the explanatory variables
- **offset** numeric vector: a vector of length equal to the number of cases

Value

The log-likelihood value of the COM-Poisson model.
COM_Poisson_Distribution

The Conway-Maxwell-Poisson (COM-Poisson) Distribution.

Description

Density, distribution function, quantile function and random generation for the Conway-Maxwell-Poisson distribution with parameter \( \mu \) and \( \nu \).

Usage

dcomp(x, mu = 1, lambda, log.p = FALSE, lambdalb = 1e-10, lambdaub = 1900, maxlambdaiter = 1000, tol = 1e-06)

pcomp(q, mu = 1, lambda, lower.tail = TRUE, log.p = FALSE, lambdalb = 1e-10, lambdaub = 1900, maxlambdaiter = 1000, tol = 1e-06)

qcomp(p, mu = 1, lambda, lower.tail = TRUE, log.p = FALSE, lambdalb = 1e-10, lambdaub = 1900, maxlambdaiter = 1000, tol = 1e-06)

rcomp(n, mu = 1, lambda, lambdalb = 1e-10, lambdaub = 1900, maxlambdaiter = 1000, tol = 1e-06)

Arguments

- \( x \), \( q \): vector of quantiles
- \( \mu \), \( \nu \): mean and dispersion parameters. Must be strictly positive.
- \( \lambda \): an alternative way than \( \mu \) to parametrize the distribution. Must be strictly positive.
- \( \log.p \): logical; if TRUE, probabilities/densities \( p \) are returned as \( \log(p) \).
- \( \lambda_{lb} \), \( \lambda_{ub} \): numeric; the lower and upper end points for the interval to be searched for \( \lambda \)(s).
- \( \text{maxlambdaiter} \): numeric: the maximum number of iterations allowed to solve for \( \lambda \)(s).
- \( \text{tol} \): numeric: the convergence threshold. A \( \lambda \) is said to satisfy the mean constraint if the absolute difference between the calculated mean and \( \mu \) is less than \( \text{tol} \).
- \( \text{lower.tail} \): logical; if TRUE (default), probabilities are \( P(X \leq x) \), otherwise, \( P(X > x) \).
- \( p \): vector of probabilities
- \( n \): number of observations. If \( \text{length}(n) > 1 \), the length is taken to be the number required.
Value

dcomp gives the density, pcomp gives the distribution function, qcomp gives the quantile function, and rcomp generates random deviates.

Invalid arguments will result in return value NaN, with a warning.

The length of the results is determined by n for rcomp, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than n are recycled to the length of the results. Only the first argument of the logical arguments are used.

Examples

dcomp(0:5, mu = 2, nu = 1.2)
pcomp(5, mu=2, nu =1.2)
p <- (1:9)/10
qcomp(p, mu = 2, nu = 0.8)
rcomp(10, mu = 2, nu = 0.7)

---

cottonbolls Cotton Bolls data set

Description

This data set gives the observed number of bolls produced by the cotton plants at five growth stages: vegetative, flower-bud, blossom, fig and cotton boll; to examine the effect of five defoliation levels (0

Usage

data(cottonbolls)

Format

A data frame with 125 observations on 4 variables.

nc number of bolls produced by two cotton plants at harvest
stages growth stage
def artificial defoliation level
def2 square of def

Source

fitted.cmp

References


Examples

```r
## For examples see example(glm.cmp)
```

---

**fitted.cmp**  
*Extract Fitted Values from a COM-Poisson Model Fit*

**Description**

An accessor function used to extract the fitted values from a 'cmp' object. `fitted.values` is an alias for `fitted`.

**Usage**

```r
## S3 method for class 'cmp'
fitted(object, ...)
```

**Arguments**

- `object`  
  an object class 'cmp' object, obtained from a call to `glm.cmp`

- `...`  
  other arguments passed to or from other methods (currently unused).

**Value**

Fitted values μ extracted from the object `object`.

**See Also**

`coef.cmp`, `residuals.cmp`, `glm.cmp`.  

**getnu**  
*Parameter Generator for nu*

**Description**

A function that use the arguments of a `glmcmp` call to generate a better initial `nu` estimate.

**Usage**

```r
getnu(param, y, xx, offset, llstart, fsscale = 1, lambdalb = 1e-10, lambdaub = 1900, maxlambdaiter = 1000, tol = 1e-06)
```

**Arguments**

- `param` numeric vector: the model coefficients & the current value of nu. It is assumed that nu is in the last position of param.
- `y` numeric vector: response variable
- `xx` numeric matrix: the explanatory variables
- `offset` numeric vector: a vector of length equal to the number of cases
- `llstart` numeric: current log-likelihood value
- `fsscale` numeric: a scaling factor (generally >1) for the relaxed fisher scoring algorithm
- `lambdalb, lambdaub` numeric: the lower and upper end points for the interval to be searched for lambda(s).
- `maxlambdaiter` numeric: the maximum number of iterations allowed to solve for lambda(s).
- `tol` numeric: the convergence threshold. A lambda is said to satisfy the mean constraint if the absolute difference between the calculated mean and a fitted values is less than tol.

**Value**

List containing the following:

- `param` the model coefficients & the updated `nu`
- `maxl` the updated log-likelihood
- `fsscale` the final scaling factor used
glm.cmp

Fit a Mean Parametrized Conway-Maxwell-Poisson Generalized Linear Model

Description

The function glm.cmp is used to fit a mean parametrized Conway-Maxwell-Poisson generalized linear model with a log-link by using Fisher Scoring iteration.

Usage

```r
glm.cmp(formula, data, offset = NULL,
        lambdalb = 1e-10, lambdaub = 1900, maxlambdaiter = 1e3, tol = 1e-6)
```

Arguments

- `formula`: an object of class 'formula': a symbolic description of the model to be fitted.
- `data`: an optional data frame containing the variables in the model
- `offset`: this can be used to specify an *a priori* known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length equal to the number of cases.
- `lambdalb, lambdaub`: numeric: the lower and upper end points for the interval to be searched for lambda(s). The default value for lambdaub should be sufficient for small to moderate size nu. If nu is large and required a larger lambdaub, the algorithm will scale up lambdaub accordingly.
- `maxlambdaiter`: numeric: the maximum number of iterations allowed to solve for lambda(s).
- `tol`: numeric: the convergence threshold. A lambda is said to satisfy the mean constraint if the absolute difference between the calculated mean and a fitted values is less than tol.

Details

Fit a mean parametrized COM-Poisson regression using maximum likelihood estimation via an iterative Fisher Scoring algorithm.

The COM-Poisson regression model is

\[ Y_i \sim CMP(\mu_i, \nu), \]

where

\[ E(Y_i) = \mu_i = \exp(x_i^T \beta), \]

and \( \nu > 0 \) is the dispersion parameter.

The fitted COM-Poisson distribution is over- or under-dispersed if \( \nu < 1 \) and \( \nu > 1 \) respectively.
Value

A fitted model object of class cmp similar to one obtained from glm or glm.nb.

The function summary (i.e., summary.cmp) can be used to obtain and print a summary of the results. The function plot (i.e., plot.cmp) can be used to produce a range of diagnostic plots.

The generic assessor functions coef (i.e., coef.cmp), logLik (i.e., logLik.cmp) fitted (i.e., fitted.cmp), nobs (i.e., nobs.cmp), AIC (i.e., AIC.cmp) and residuals (i.e., residuals.cmp) can be used to extract various useful features of the value returned by glm.cmp.

An object class 'glm.cmp' is a list containing at least the following components:

- coefficients: a named vector of coefficients
- se_beta: approximate standard errors (using observed rather than expected information) for coefficients
- residuals: the response residuals (i.e., observed-fitted)
- fitted.values: the fitted mean values
- rank: the numeric rank of the fitted linear model
- linear.predictors: the linear fit on log scale
- df.residuals: the residuals degrees of freedom
- df.null: the residual degrees of freedom for the null model
- null.deviance: The deviance for the null model. The null model will include only the intercept.
- y: the y vector used.
- x: the model matrix
- model: the model frame
- call: the matched call
- formula: the formula supplied
- terms: the terms object used
- data: the data argument
- offset: the offset vector used
- lambdaub: the final lambdaub used

References


See Also

summary.cmp, plot.cmp, fitted.cmp and residuals.cmp.
is.wholenumber

Examples

### Huang (2017) Page 368--370: Overdispersed Attendance data

data(attendance)
M.attendance <- glm.cmp(daysabs~ gender+math+prog, data=attendance)
M.attendance
summary(M.attendance)

### Huang (2017) Page 371--372: Underdispersed Takeover Bids data

data(takeoverbids)
M.bids <- glm.cmp(numbids ~ leglrest + rearest + finrest + whtknght
   + bidprem + insthold + size + sizesq + regulatn, data=takeoverbids)
M.bids
summary(M.bids)

### Huang (2017) Page 373--375: Underdispersed Cotton bolls data

### Model fitting for predictor V

data(cottonbolls)
M.bolls <- glm.cmp(nc~ 1+stages:;def+stages:;def2, data= cottonbolls)
M.bolls
summary(M.bolls)

is.wholenumber [Test for a whole number]

Description

Test for integer/whole number vector

Usage

is.wholenumber(x, tol = .Machine$double.eps^0.5)

Arguments

x numeric vector to be tested
tol numeric; precision level
**logLik.cmp**

*Extract the (Maximized) Log-Likelihood from a COM-Poisson Model Fit*

---

**Description**

An accessor function used to extract the (maximized) log-likelihood from a 'cmp' object.

**Usage**

```r
## S3 method for class 'cmp'
logLik(object, ...)

## S3 method for class 'logLik.cmp'
print(x, ...)
```

**Arguments**

- `object`: an object of class 'cmp' object, obtained from a call to `glm.cmp`
- `...`: other arguments passed to or from other methods (currently unused).
- `x`: an object of class 'logLik.cmp', obtained from a call to `logLik.cmp`.

**See Also**

`coef.cmp`, `fitted.cmp`, `glm.cmp`

---

**LRTnu**

*Likelihood Ratio Test for nu = 1 of a COM-Poisson model*

---

**Description**

Perform a likelihood ratio chi-squared test for nu = 1 of a COM-Poisson model. The test statistics is calculated as $2*(\text{llik} - \text{llik}_0)$ where `llik` and `llik_0` are the log-likelihood of a COM-Poisson and Poisson model respectively. The test statistic has 1 degrees of freedom.

**Usage**

```r
LRTnu(object, digits = 3)
```

**Arguments**

- `object`: an object class 'cmp', obtained from a call to `glm.cmp`
- `digits`: numeric; minimum number of significant digits to be used for most numbers.
References


Examples

```r
data(takeoverbids)
M.bids <- glm.cmp(numbids ~ leglrest + rearest + finrest + whtknight
                   + bidprem + insthold + size + sizesq + regulatn, data=takeoverbids)
LRTnu(M.bids)
```

---

**model.frame.cmp**

*Extract the Model Frame from a COM-Poisson Model Fit*

**Description**

An accessor function used to extract the model frame from a `cmp` object.

**Usage**

```r
# S3 method for class 'cmp'
model.frame(formula, ...)
```

**Arguments**

- `formula` an object class ‘cmp’ object, obtained from a call to `glm.cmp`
- `...` other arguments passed to or from other methods (currently unused).

**Value**

The method will return the saved `data.frame` used when fitting the cmp model.

**See Also**

`coef.cmp`, `residuals.cmp`, `glm.cmp`. 
**Description**

An accessor function used to extract the number of observation from a 'cmp' object.

**Usage**

```r
## S3 method for class 'cmp'
nobs(object, ...)
```

**Arguments**

- `object`: an object class 'cmp' object, obtained from a call to `glm.cmp`
- `...`: other arguments passed to or from other methods (currently unused).

**Value**

The number of observations extracted from the object `object`.

**See Also**

`coef.cmp, fitted.cmp, glm.cmp`

---

**nrPIT**

*Non-randomized Probability Integral Transform*

**Description**

Functions to produce the non-randomized probability integral transform (PIT) to check the adequacy of the distributional assumption of the COM-Poisson model. The majority of the code and descriptions are taken from Dunsmuir and Scott (2015).

**Usage**

```r
compPredProb(object)
compPIT(object, bins = 10)
```

**Arguments**

- `object`: an object class "cmp", obtained from a call to `glm.cmp`.
- `bins`: numeric; the number of bins shown in the PIT histogram or the PIT Q-Q plot.
Details

These functions are used to obtain the predictive probabilities and the probability integral transform for a fitted COM-Poisson model. The majority of the code and descriptions are taken from Dunsmuir and Scott (2015).

Value

comppredprob returns a list with values:

upper the predictive cumulative probabilities used as the upper bound for computing the non-randomized PIT.

lower the predictive cumulative probabilities used as the upper bound for computing the non-randomized PIT.

comppit returns a list with values:

conditionalPIT the conditional probability integral transformation given the observed counts.

PIT the probability integral transformation.

References


Examples

```r
data(takeoverbids)
M.bids <- glm.cmpprob(numbids ~ leglrest + rearest + finrest + whtknght
+ bidprem + insthold + size + sizesq + regulatn, data=takeoverbids)
comppredProb(M.bids)
comppIT(M.bids)
```

Description

Two plots for the non-randomized PIT are currently available for checking the distributional assumption of the fitted CMP model: the PIT histogram, and the uniform Q-Q plot for PIT.

Usage

```r
histcomppit(object, bins = 10, line = TRUE, colLine = ”red”,
    colHist = ”royal blue”, lwdLine = 2, main = NULL, ...)

qqcomppit(object, bins = 10, col1 = ”red”, col2 = ”black”,
    lty1 = 1, lty2 = 2, type = ”l”, main = NULL, ...)
```
Arguments

- object: an object class "cmp", obtained from a call to glm.cmp.
- bins: numeric; the number of bins shown in the PIT histogram or the PIT Q-Q plot.
- line: logical; if TRUE (default), the line for displaying the standard uniform distribution will be shown for the purpose of comparison.
- colLine: numeric or character; the colour of the line for comparison in PIT histogram.
- colHist: numeric or character; the colour of the histogram for PIT.
- lwdLine: numeric; the line widths for the comparison line in PIT histogram.
- main: character string; a main title for the plot.
- ... other arguments passed to plot.default and plot.ts.
- col1: numeric or character; the colour of the sample uniform Q-Q plot in PIT.
- col2: numeric or character; the colour of the theoretical uniform Q-Q plot in PIT.
- lty1: integer or character string: the line types for the sample uniform Q-Q plot in PIT, see par(lty = ).
- lty2: an integer or character string: the line types for the theoretical uniform Q-Q plot in PIT, see par(lty = ).
- type: 1-character string; the type of plot for the sample uniform Q-Q plot in PIT.

Details

The histogram and the Q-Q plot are used to compare the fitted profile with a standard uniform distribution. If they match relatively well, it means the CMP distribution is appropriate for the data.

References


Examples

```r
## For examples see example(plot.cmp)
```

---

### plot.cmp

**Plot Diagnostic for a glm.cmp Object**

Description

Eight plots (selectable by which) are currently available: a plot of deviance residuals against fitted values, a non-randomized PIT histogram, a uniform Q-Q plot for non-randomized PIT, a histogram of the normal randomized residuals, a Q-Q plot of the normal randomized residuals, a Scale-Location plot of sqrt(1 residual) against fitted values a plot of Cook's distances versus row labels a plot of pearson residuals against leverage. By default, four plots (number 1, 2, 6, and 8 from this list of plots) are provided.
Usage

```r
## S3 method for class 'cmp'
plot(x, which = c(1L, 2L, 6L, 8L),
     ask = prod(par("mfcol")) < length(which) && dev.interactive(),
     bins = 10, ...)```

Arguments

- `x`: an object class `cmp` object, obtained from a call to `glm.cmp`
- `which`: if a subset of plots is required, specify a subset of the numbers 1:8. See 'Details' below.
- `ask`: logical; if `TRUE`, the user is asked before each plot.
- `bins`: numeric; the number of bins shown in the PIT histogram or the PIT Q-Q plot.
- `...`: other arguments passed to or from other methods (currently unused).

Details

The 'Scale-Location' plot, also called 'Spread-Location' plot, takes the square root of the absolute standardized deviance residuals (\(\sqrt{|E|}\)) in order to diminish skewness is much less skewed than \(|E|\) for Gaussian zero-mean \(E\).

The 'Scale-Location' plot uses the standardized deviance residuals while the Residual-Leverage plot uses the standardized pearson residuals. They are given as \(R[i]/\sqrt{1-h_{ii}}\) where \(h_{ii}\) are the diagonal entries of the hat matrix.

The Residuals-Leverage plot shows contours of equal Cook’s distance for values of 0.5 and 1.

There are two plots based on the non-randomized probability integral transformation (PIT) using `compPIT`. These are a histogram and a uniform Q-Q plot. If the model assumption is appropriate, these plots should reflect a sample obtained from a uniform distribution.

There are also two plots based on the normal randomized residuals calculated using `compnormRandPIT`. These are a histogram and a normal Q-Q plot. If the model assumption is appropriate, these plots should reflect a sample obtained from a normal distribution.

See Also

- `compPIT`, `compnormRandPIT`, `glm.cmp`

Examples

```r
data(takeoverbids)
M.bids <- glm.cmp(numbids ~ legrest + rearest + finrest + whtknght + bidprem + insthold + size + sizesq + regulatn, data=takeoverbids)

## The default plots are shown
plot(M.bids)

## The plots for the non-randomized PIT
# plot(M.bids, which = c(2,3))```
Description

This is a function for obtaining predictions and optionally estimates standard errors of those prediction from a fitted COM-Poisson regression object.

Usage

```r
# S3 method for class 'cmp'
predict(object, newdata = NULL, se.fit = FALSE,
         type = c("link", "response"), ...)
```

Arguments

- `object`: an object class 'cmp', obtained from a call to `glm.cmp`.
- `newdata`: optionally, a data frame in which to look for variables with which to predict. If omitted, the fitted linear predictors are used.
- `se.fit`: logical; indicating if standard errors are required.
- `type`: the type of prediction required. The default is 'link' which is the scale of the linear predictor i.e., a log scale; the alternative 'response' is on the scale of the response variable. The value of this argument can be abbreviated.
- `...`: other arguments passed to or from other methods (currently unused).

Details

If `newdata` is omitted the predictions are based on the data used for the fit.

Value

- If `se.fit = FALSE`, a vector of predictions.
- If `se.fit = TRUE`, a list with components
  - `fit`: Predictions, as for `se.fit = FALSE`.
  - `se.fit`: Estimated standard errors.

Examples

```r
data(takeoverbids)
M.bids <- glm.cmp(numbids ~ leglrest + rearest + finrest + whtknght
                  + bidprem + insthold + size + sizesq + regulatn, data=takeoverbids)
predict(M.bids)
predict(M.bids, type = "response")
predict(M.bids, se.fit=TRUE, type="response")
```
newdataframe <- data.frame(bidprem = 1L, finrest = 0L, insthold = 0.05L, 
  leglrest = 0L, rearest = 1L, regultn = 0L, size = 0.1L, whtknght = 1L, 
  sizesq = 1^2)
predict(M.bids, se.fit=TRUE, newdata = newdataframe, type="response")
Arguments

object an object class 'cmp', obtained from a call to glm.cmp.
type the type of residuals which should be returned. The alternatives are: 'deviance' (default), 'pearson' and 'response'. Can be abbreviated.
... other arguments passed to or from other methods (currently unused).

Value

Residuals extracted from the object object.

See Also

coef.cmp, fitted.cmp, glm.cmp

rPIT Random Normal Probability Integral Transform

Description

A function to create the normal conditional (randomized) quantile residuals. The majority of the code and descriptions are taken from Dunsmuir and Scott (2015).

Usage

compnormRandPIT(object)

Arguments

object an object class "cmp", obtained from a call to glm.cmp.

Details

The function compPredProb produces the non-randomized probability integral transform(PIT). It returns estimates of the cumulative predictive probabilities as upper and lower bounds of a collection of intervals. If the model is correct, a histogram drawn using these estimated probabilities should resemble a histogram obtained from a sample from the uniform distribution.

This function aims to produce observations which instead resemble a sample from a normal distribution. Such a sample can then be examined by the usual tools for checking normality, such as histograms and normal Q-Q plots.

For each of the intervals produced by compPredProb, a random uniform observation is generated, which is then converted to a normal observation by applying the inverse standard normal distribution function (using qnorm). The vector of these values is returned by the function in the list element rt. In addition non-random observations which should appear similar to a sample from a normal distribution are obtained by applying qnorm to the mid-points of the predictive distribution intervals. The vector of these values is returned by the function in the list element rtMid.
**Value**

A list consisting of two elements:

- `rt` the normal conditional randomized quantile residuals
- `rdMid` the midpoints of the predictive probability intervals

**References**


**Examples**

```r
data(takeoverbids)
M.bids <- glm cmp(numbids ~ leglrest + rearest + finrest + whtnght
+ bidprem + insthold + size + sizesq + regulatn, data=takeoverbids)
compnormRandPIT(M.bids)
```

---

**summary.cmp**

Summary of COM-Poisson Model Fit

**Description**

summary method for class cmp.

**Usage**

```r
## S3 method for class 'cmp'
summary(object, digits = max(3L, getOption("digits") - 3L),
         ...)
```

**Arguments**

- `object` an object class 'cmp', obtained from a call to glm.cmp.
- `digits` numeric; minimum number of significant digits to be used for most numbers.
- `...` other arguments passed to or from other methods (currently unused).

**Details**

summaryglm tries to be smart about formatting the coefficients, standard errors and gives 'significance starts'. The coefficients component of the result gives the estimated coefficients and their estimated standard errors, together with their ratio. This third column is labelled as Z value as the dispersion is fixed for this family. A forth column gives the two-tailed p-value corresponding to Z value based on the asymptotic Normal reference distribition.
See Also

`coef.cmp`, `fitted.cmp`, `glm.cmp`.

Examples

```r
## For examples see example(glm.cmp)
```

---

**takeoverbids**

*Takeover Bids data set*

Description

This data set gives the number of bids received by 126 US firms that were successful targets of tender offers during the period 1978–1985, along with some explanatory variables on the defensive actions taken by management of target firm, firm-specific characteristics and intervention taken by federal regulators. The `takeoverbids` data frame has 126 observations on 14 variables. The descriptions below are taken from Sáez-Castillo and Conde-Sánchez (2013).

Usage

```r
data(takeoverbids)
```

Format

A data frame with 126 observations on 14 variables.

- **bidprem**: bid price divided by price 14 working days before bid
- **docno**: doc no
- **finrest**: indicator variable for proposed change in ownership structure
- **insthold**: percentage of stock held by institutions
- **leglrest**: indicator variable for legal defence by lawsuit
- **numbids**: number of bids received after the initial bid
- **obs**: Identifier
- **rearest**: indicator variable for proposed changes in asset structure
- **regulatn**: indicator variable for Department of Justice intervention
- **size**: total book value of assets in billions of dollars
- **takeover**: Indicator. 1 if the company was being taken over
- **weeks**: time in weeks between the initial and final offers
- **whltkght**: indicator variable for management invitation for friendly third-party bid
- **sizesq**: book value squared

Source

References


Examples

```r
## For examples see example(glm.cmp)
```

---

**update.cmp**

*Update and Re-fit a COM-Poisson Model*

**Description**

update (i.e., `update.cmp`) will update and (by-default) re-fit a model. It is identical to `update` in the `stats` package.

**Usage**

```r
## S3 method for class 'cmp'
update(object, formula, ..., evaluate = TRUE)
```

**Arguments**

- **object**: an object class `cmp`, obtained from a call to `glm.cmp`.
- **formula**: changes to the existing formula in `object` – see `update.formula` for details
- **...**: other arguments passed to or from other methods (currently unused).
- **evaluate**: logical; if `TRUE` evaluate the new call otherwise simply return the call

**See Also**

`glm.cmp`, `update.formula`, `cmplrtest`.

**Examples**

```r
data(takeoverbids)

## Fit full model
M.bids.full <- glm.cmp(numbids ~ leglrest + rearest + finrest + whtnght + bidprem + insthold + size + sizesq + regulatn, data=takeoverbids)
M.bids.full
```
Calculate the Normalizing Constant for COM-Poisson distribution

Description
A function to approximate the normalizing constant for COM-Poisson distributions via truncation. The standard COM-Poisson parametrization is being used here. Notice that the sum is hard coded to truncate at 100 so the approximation will be quite bad if the COM-Poisson has a large rate or mean.

Usage
Z(lambda, nu)

Arguments
lambda rate parameter, strictly positive
nu diepsersoin parameter, strictly positive
Index

*Topic datasets
  attendance, 3
cottonbolls, 10
takeoverbids, 26
*Topic package
  mpcmp-package, 2

AIC.cmp, 3, 14
attendance, 3

CBIND, 4
cmpLRtest, 5, 27
coeff.cmp, 6, 11, 14, 16–18, 23, 24, 26
COM_Poisson_Distribution, 9
comp_expected_values, 6
comp_lambdas, 7
comp_means_logfactorialy
  (COMP_Expected_Values), 6
comp_meanc_lfactorialy
  (COMP_Expected_Values), 6
comp_means
  (COMP_Expected_Values), 6
comp_mu_loglik, 8
comp_variances
  (COMP_Expected_Values), 6
comp_variances_logfactorialy
  (COMP_Expected_Values), 6
compnormRandPIT, 21
compnormRandPIT (rPIT), 24
compPIT, 21
compPIT (nrPIT), 18
compPredProb (nrPIT), 18
cottonbolls, 10

data.frame, 17
dcomp (COM_Poisson_Distribution), 9

fitted.cmp, 6, 11, 14, 16, 18, 23, 24, 26

getnu, 12
glm.cmp, 3, 5, 6, 11, 13, 16–18, 21, 23, 24, 26.

histcompPIT (PIT_Plot), 19
is.wholenumber, 15
logLik.cmp, 3, 14, 16
LRTnus, 16

model.frame.cmp, 17
mpcmp (mpcmp-package), 2
mpcmp-package, 2

nobs.cmp, 3, 14, 18
nrPIT, 18

pcomp (COM_Poisson_Distribution), 9
PIT_Plot, 19
plot.cmp, 14, 20
predict.cmp, 22
print.cmp, 23
print.logLik.cmp (logLik.cmp), 16

qcomp (COM_Poisson_Distribution), 9
qqcompPIT (PIT_Plot), 19

rcomp (COM_Poisson_Distribution), 9
residuals.cmp, 6, 11, 14, 17, 23
rPIT, 24

summary.cmp, 14, 23, 25
takeoverbids, 26

update.cmp, 5, 27
update.formula, 27

Z, 28