

# Package ‘MRDCr’

July 21, 2016

**Title** Modelos de Regressão para Dados de Contagem

**Version** 0.0-2

**Date** 2016-07-21

**Description** Esse é um pacote com o material do Curso "Modelos de Regressão para Dados de Contagem" oferecido pelos autores na 61 RBRAS (rbras2016.ufba.br) que aconteceu entre 23 e 25 de Maio em Salvador, Bahia.

O objetivo desse Curso é apresentar diferentes modelos de regressão para análise de dados de contagem, discutir suas principais propriedades e ilustrar suas aplicações em dados reais por meio do software R. Dentre os modelos considerados estão o modelo Poisson, algumas extensões para acomodação de super (ou sub) dispersão (Quasi-Poisson, COM-Poisson, Count-Gamma, Binomial-Negativo), modelos para dados inflacionados de zeros e com a inclusão de efeitos aleatórios.

Neste pacote estão disponíveis as vignettes que mostram como coordenar funções do R para análise dos dados, os slides do Curso, as funções e alguns dos dados utilizados nos estudos de caso.

**License** GPL-3

**URL** <https://gitlab.c3sl.ufpr.br/leg/MRDCr>

**BugReports** <https://gitlab.c3sl.ufpr.br/leg/MRDCr/issues>

**LazyData** true

**Encoding** UTF-8

**Depends** R (>= 3.2.3), bbmle

**Imports** lattice, latticeExtra, doBy, multcomp

**Suggests** MASS, car, corrplot, hnp, pscl, effects, lmtest, boot, RColorBrewer, lme4, sandwich, plyr, reshape, reshape2, knitr, rmarkdown, testthat, methods

**VignetteBuilder** knitr

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Author** Walmes Marques Zeviani [cre, aut],  
Eduardo Elias Ribeiro Junior [aut],  
Cesar Augusto Taconeli [aut]

**Maintainer** Walmes Marques Zeviani <walmes@ufpr.br>

## R topics documented:

apc . . . . .	3
calc_mean_cmp . . . . .	4
calc_mean_gcnt . . . . .	5
calc_var_cmp . . . . .	6
cambras . . . . .	6
capdesfo . . . . .	8
capmosca . . . . .	9
cmp . . . . .	11
confterm . . . . .	12
convergencez . . . . .	14
dcmp . . . . .	15
dgcnt . . . . .	16
dpgnz . . . . .	17
gcnt . . . . .	19
led . . . . .	20
llcmp . . . . .	21
llgcnt . . . . .	23
llpgnz . . . . .	25
MRDCr . . . . .	27
nematoide . . . . .	27
ninfas . . . . .	28
panel.beeswarm . . . . .	30
panel.cbH . . . . .	31
panel.groups.segplot . . . . .	33
peixe . . . . .	34
pgnz . . . . .	35
postura . . . . .	36
seguros . . . . .	37
soja . . . . .	38
<b>Index</b>	<b>40</b>

**Description**

Essa função retorna os contrastes de Tukey entre médias a partir da matriz que define as funções lineares dos coeficientes do modelo para estimar a média de mínimos quadrados. Essas matrizes podem ser utilizadas para fazer contrastes entre médias por meio da função `multcomp::glht()`.

**Usage**

```
apc(lfm, lev = NULL)
```

**Arguments**

<code>lfm</code>	Uma matriz de dimensões $k \times p$ onde cada linha tem os coeficientes correspondentes para estimar uma média. Essas matrizes são facilmente obtidas com usando a função <code>doBy::LSmatrix()</code> .
<code>lev</code>	Um vetor com os nomes dos níveis do fator para o qual correspondem as linhas da matriz usada em <code>lfm</code> . Portanto, o número de elementos do vetor deve ser igual ao número de linhas da matriz. O valor default é <code>NULL</code> e então é usado o <code>rownames()</code> da matriz. Se <code>rownames()</code> for <code>NULL</code> , uma sequência de números começando em 1 é utilizada para representar cada linha.

**Value**

Uma matriz  $r \times p$  cujas linhas estimam o contraste entre cada possível par de médias, portanto  $r$  is  $\binom{k}{2}$ .

**Author(s)**

Walmes Zeviani baseado na lista de discussão R-help.

**See Also**

[LSmatrix](#).

**Examples**

```
X <- diag(3)
rownames(X)
apc(X)

rownames(X) <- LETTERS[nrow(X):1]
apc(X)

apc(X, lev = LETTERS[1:nrow(X)])
```

```

#-----
# Usando para estimar as médias e contrastes.

xtabs(~tension + wool, data = warpbreaks)

warpbreaks <- transform(warpbreaks,
                        trt = interaction(tension, wool))

m0 <- lm(log(breaks) ~ trt, data = warpbreaks)
anova(m0)

library(doBy)

L <- LSmatrix(m0, effect = "trt")
L

K <- apc(L)
K

library(multcomp)

glht(m0, linfct = L)
glht(m0, linfct = K)

aggregate(cbind(log(breaks)) ~ trt, data = warpbreaks, FUN = mean)

```

---

calc\_mean\_cmp

*NAConway-Maxwell-Poisson*


---

### Description

Função para calcular a média do tipo  $E(Y) = \mu = \sum y \cdot \Pr(y)$  para uma variável aleatória COM-Poisson a partir dos parâmetros  $\lambda > 0$  e  $\nu \geq 0$ .

### Usage

```
calc_mean_cmp(lambda, nu, sumto, tol = 1e-05)
```

### Arguments

lambda	Valor do parâmetro $\lambda$ da distribuição COM-Poisson. Quando $\nu = 1$ , o parâmetro $\lambda = E(Y)$ é a média.
nu	Valor do parâmetro $\nu$ da distribuição COM-Poisson.
sumto	Número de incrementos a serem considerados para a cálculo da constante normalizadora $Z$ .
tol	Tolerância para interromper a procura pelo valor de $y_{\max}$ , valor cuja probabilidade correspondente é inferior a $tol$ , para valores os valores de $lambda$ e $nu$ informados.

**Value**

Um vetor de tamanho igual ao do maior vetor, lambda ou nu com os valores correspondentes de  $\mu$ .

**Author(s)**

Eduardo E. R. Junior, <edujrrib@gmail.com>

---

calc\_mean\_gcnt

*Calcula o Valor Esperado para a Distribuicao Gamma Count*

---

**Description**

Função para calcular a média do tipo  $E(Y) = \mu = \sum y \cdot \Pr(y)$  para uma variável aleatória Gamma Count a partir dos parâmetros  $\lambda > 0$  e  $\alpha > 0$ .

**Usage**

```
calc_mean_gcnt(lambda, alpha, tol = 1e-05)
```

**Arguments**

lambda	Parâmetro de locação da distribuição Gamma Count, $\lambda > 0$ . Quando $\alpha = 1$ , o parâmetro $\lambda = E(Y)$ é a média.
alpha	Parâmetro de dispersão da distribuição Gamma Count, $\alpha > 0$ .
tol	Tolerância para interromper a procura pelo valor de ymax, valor cuja probabilidade correspondente é inferior a tol, para valores os valores de lambda e alpha informados.

**Value**

Um vetor de tamanho igual ao do maior vetor, lambda ou alpha com os valores correspondentes de  $\mu$ .

**Author(s)**

Walmes Zeviani, <walmes@ufpr.br>.

---

 calc\_var\_cmp

*NAConway-Maxwell-Poisson*


---

**Description**

Função para calcular a variância do tipo  $V(Y) = E(Y^2) - E^2(Y) = \sum y^2 \cdot \text{Pr}(y) - (\sum y \cdot \text{Pr}(y))^2$  para uma variável aleatória COM-Poisson a partir dos parâmetros  $\lambda > 0$  e  $\nu \geq 0$ .

**Usage**

```
calc_var_cmp(lambda, nu, sumto, tol = 1e-05)
```

**Arguments**

lambda	Valor do parâmetro $\lambda$ da distribuição COM-Poisson. Quando $\nu = 1$ , o parâmetro $\lambda = E(Y)$ é a média.
nu	Valor do parâmetro $\nu$ da distribuição COM-Poisson.
sumto	Número de incrementos a serem considerados para a cálculo da constante normalizadora $Z$ .
tol	Tolerância para interromper a procura pelo valor de $y_{\max}$ , valor cuja probabilidade correspondente é inferior a $tol$ , para valores os valores de $lambda$ e $nu$ informados.

**Value**

Um vetor de tamanho igual ao do maior vetor,  $lambda$  ou  $nu$  com os valores correspondentes de  $\mu$ .

**Author(s)**

Eduardo E. R. Junior, <edujrrib@gmail.com>

---

 cambras

*Resultados do Campeonato Brasileiro de 2010*


---

**Description**

Dados sobre o número de gols dos times mandantes e desafiante de todas as partidas do Campeonato Brasileiro de 2010.

**Format**

Um data.frame com 380 observações e 5 variáveis.

rod Inteiro que identifica o número da rodada.

home Fator que identifica o time mandante da partida, aquele que jogou em casa.

away Fator que identifica o time desafiante da partida, aquele que jogou fora de casa.

h Número de gols feitos pelo time mandante na partida.

a Número de gols feitos pelo time desafiante na partida.

**Examples**

```
# Resultados finais na internet.
# browseURL(paste0("http://esportes.terra.com.br/futebol/",
#                 "brasileiro/2010/noticias/0,,OI4339585-EI15413",
#                 ",00-Classificacao+e+Jogos+Serie+A.html"))

Xh <- model.matrix(~-1 + home, cambras)
Xa <- model.matrix(~-1 + away, cambras)

Xha <- Xh - Xa
cambras[1:5, c("home", "away")]
print(as.table(t(Xha[1:5, ])), zero.print = ".")

gsgc <- cbind(scored = t(Xh) %*% cambras$h + t(Xa) %*% cambras$a,
              conceded = t(Xa) %*% cambras$h + t(Xh) %*% cambras$a)
colnames(gsgc) <- c("gScored", "gConced")
gsgc

# Pontos em cada partida (empate = 1, vitória = 3).
pts <- with(cambras, {
  d <- h - a
  draw <- d == 0
  winH <- d > 0
  winA <- !(draw | winH)
  x <- cbind(h = winH * 3 + draw, a = winA * 3 + draw)
  return(x)
})

tableIn <- function(x) {
  tb <- table(x)
  f <- rep(0, 3)
  names(f) <- c(0, 1, 3)
  f[names(tb)] <- tb
  return(f)
}

# Derrotas, empates e vitórias.
ldw <-
  do.call(rbind, lapply(tapply(pts[, 1], cambras$home,
                              FUN = tableIn),
```

```

      FUN = as.vector)) +
do.call(rbind, lapply(tapply(pts[, 2], cambras$away,
      FUN = tableIn),
      FUN = as.vector))
colnames(ldw) <- c("Lose", "Draw", "Win")
ldw

# Tabela completa de 2010.
pl10 <- t(Xh) %*% pts[, "h"] + t(Xa) %*% pts[, "a"]
pl10 <- cbind(team = levels(cambras$home), data.frame(pts = pl10))
pl10 <- cbind(pl10, ldw[, 3:1], gsgc)
pl10 <- plyr::arrange(pl10, -pts)
pl10$pos <- rank(-pl10$pts)
pl10

```

---

 capdesfo

*Capulhos de Algodao em Funcao de Desfolha Artificial*


---

### Description

Experimento conduzido sob delineamento inteiramente casualizado com 5 repetições em casa de vegetação com plantas de algodão *Gossypium hirsutum* submetidas à diferentes níveis de desfolha artificial de remoção foliar, em combinação com o estágio fenológico no qual a desfolha foi aplicada. A unidade experimental foi um vaso com duas plantas onde avaliou-se o número de capulhos produzidos ao final da ciclo cultura.

### Format

Um codedata.frame com 125 observações e 4 variáveis

est Um fator categórico com 5 níveis que representa o estágio fenológico da planta durante a aplicação da desfolha.

des Um fator numérico com 5 níveis que representa o nível de desfolha artificial (percentual da área da folha removida com tesoura) aplicada a todas as folhas na planta.

rept Inteiro que representa cada unidade experimental.

ncap Inteiro que representa o número de capulhos de algodão produzidos ao final da ciclo cultura.

### References

Silva, A. M., Degrande, P. E., Suekane, R., Fernandes, M. G., & Zeviani, W. M. (2012). Impacto de diferentes níveis de desfolha artificial nos estádios fenológicos do algodoeiro. *Revista de Ciências Agrárias*, 35(1), 163–172.

Zeviani, W. M., Ribeiro, P. J., Bonat, W. H., Shimakura, S. E., & Muniz, J. A. (2014). The Gamma-count distribution in the analysis of experimental underdispersed data. *Journal of Applied Statistics*, 41(12), 1–11. <http://doi.org/10.1080/02664763.2014.922168>



## Examples

```

data(capdesfo)
str(capdesfo)

library(lattice)

xyplot(ncap ~ des | est,
       data = capdesfo,
       layout = c(NA, 2),
       type = c("p", "smooth"),
       xlab = "Níveis de desfolha artificial",
       ylab = "Número de capulhos produzidos",
       xlim = extendrange(c(0:1), f = 0.15),
       jitter.x = TRUE,
       grid = TRUE)

# Média e variância amostral para cada unidade experimental
(mv <- aggregate(ncap ~ est + des, data = capdesfo,
                 FUN = function(x) c(mean = mean(x), var = var(x))))
xlim <- ylim <- extendrange(c(mv$ncap), f = 0.05)

# Evidência de subdispersão
xyplot(ncap[, "var"] ~ ncap[, "mean"],
       data = mv,
       xlim = xlim,
       ylim = ylim,
       ylab = "Variância Amostral",
       xlab = "Média Amostral",
       panel = function(x, y) {
         panel.xyplot(x, y, type = c("p", "r"), grid = TRUE)
         panel.abline(a = 0, b = 1, lty = 2)
       })

```

---

 capmosca

*Capulhos de Algodao em Funcao da Exposicao a Mosca Branca*


---

## Description

Experimento conduzido na Universidade Federal da Grande Dourados (UFGD) em 2007, cujo objetivo foi avaliar os impactos da exposição de plantas à alta infestação de Mosca-Branca *Bemisia tabaci* em componentes de produção do algodão. No experimento, plantas de algodão foram expostas à alta infestação da praga por períodos diferentes e ao final do experimento avaliou-se o número de capulhos produzidos, o número de estruturas reprodutivas, o número de nós, a altura da planta e o peso dos capulhos por vaso. A condução do estudo deu-se via delineamento inteiramente casualizado com 5 vasos, contendo duas plantas, para cada período de exposição.

## Format

Um data.frame com 60 observações e 8 variáveis.

dexp Inteiro com 6 valores que representa os dias de exposição à alta infestação de Mosca-Branca.

vaso Fator que indica o vaso no qual foram mensurados os componentes de produção do algodão.

planta Fator que indica a planta na qual foram mensurados os componentes de produção do algodão.

alt Altura da planta, mensurada em centímetros.

pesocap Peso dos capulhos de algodão, mesurado para cada vaso (que contém duas plantas). No data.frame somente a primeira planta do vaso contém a observação do peso.

nerep Contagem do número de estruturas reprodutivas da planta.

ncap Contagem do número de capulhos produzidos.

nnos Contagem do número de nós da planta.

## Examples

```
data(capmosca)
str(capmosca)

library(lattice)

# Número de capulhos produzidos por vaso
da <- aggregate(ncap ~ vaso + dexp, data = capmosca, FUN = sum)
xyplot(ncap ~ dexp,
       data = da,
       jitter.x = TRUE,
       type = c("p", "g", "smooth"))

# Número de capulhos produzidos por planta
xyplot(ncap ~ dexp, groups = planta,
       data = capmosca,
       jitter.x = TRUE,
       type = c("p", "g", "smooth"))

# Número de estruturas reprodutivas da planta
xyplot(nerep ~ dexp, groups = planta,
       data = capmosca,
       jitter.x = TRUE,
       type = c("p", "g", "smooth"))

# Número de nós da planta
xyplot(nnos ~ dexp, groups = planta,
       data = capmosca,
       jitter.x = TRUE,
       type = c("p", "g", "smooth"))
```

## Description

Estima os parâmetros de um modelo COM-Poisson pela otimização da função de log-verossimilhança definida em `llcmp`. A sintaxe assemelha-se com a função `glm` (Generalized Linear Models).

## Usage

```
cmp(formula, data, start = NULL, sumto = NULL, ...)
```

## Arguments

<code>formula</code>	Um objeto da classe <code>formula</code> .
<code>data</code>	Um objeto de classe <code>data.frame</code> , que contém as variáveis descritas na <code>formula</code> .
<code>start</code>	Um vetor nomeado com os valores iniciais para os parâmetros do modelo necessários para o início do procedimento de estimação. Se <code>NULL</code> as estimativas de um modelo log-linear Poisson, com $\phi = 0$ , são utilizadas como valores iniciais, pois uma chamada da <code>glm.fit</code> é feita internamente para obtê-los. O parâmetro $\phi$ deve ser o primeiro elemento do vetor. Os restantes devem estar na correspondente às colunas da matriz gerada pelo argumento <code>formula</code> .
<code>sumto</code>	Número de incrementos a serem considerados para a soma da constante normalizadora. Como padrão, <code>NULL</code> o número de incrementos é o valor inteiro de $(\max y)^{1.2}$ , porém ressalta-se que este valor padrão não é o ideal. Uma avaliação da escolha desse argumento, pós ajuste pode ser realizada via <code>convergencez</code> .
<code>...</code>	Argumentos opcionais do framework de maximização numérica <code>mle2</code> .

## Value

Um objeto de classe `mle2`, retornado da função de `mle2`, usada para ajuste de modelos por máxima verossimilhança.

## Author(s)

Eduardo E. R. Junior, <edujrrib@gmail.com>

## Examples

```
str(capdesfo)
m0 <- cmp(ncap ~ est + (des + I(des^2)), data = capdesfo, sumto = 40)
m1 <- cmp(ncap ~ est * (des + I(des^2)), data = capdesfo, sumto = 40)

convergencez(m0)
convergencez(m1)

library(bbmle)
```

```

anova(m0, m1)
summary(m1)

## Not run:
plot(profile(m1, which = "phi"))

## End(Not run)

```

---

 confterm

---

*Resfriamento da Cobertura de Aviários na Mortalidade de Aves*


---

### Description

Resultados de um experimento que estudou o efeito do sistema de resfriamento de cobertura de aviários sob a mortalidade das aves e o conforto térmico. Havia 4 aviários disponíveis e por sorteio, em dois deles foi instalado o sistema de resfriamento de cobertura, cuja finalidade é diminuir a temperatura nas horas mais quentes do dia para evitar morte de aves por calor. Nesse experimento, o sistema só foi utilizado a partir dos 21 dias de idade. A cada dia foi contado o número de aves encontradas mortas no aviário.

Fora dos galpões, um sistema de monitoramento das variáveis ambientais registrou, em intervalos de 1 hora dos 21 aos 39 dias de idade, as variáveis para que fossem determinados: a entalpia específica do ar (H), a carga térmica de radiação (CTR) e o índice de temperatura de globo negro e umidade (ITGU). Essas variáveis tem a finalidade de explicar a variação da mortalidade das aves nos sistemas de resfriamento ao longo dos dias.

### Format

confterm é um `data.frame` com 176 observações e 4 variáveis, em que

`resfr` Variável que indica se o galpão tem ou não sistema de resfriamento de cobertura (C = com resfriamento de cobertura, S = sem resfriamento de cobertura).

`idade` Idade das aves, em dias após o alojamento.

`galp` Variável que indica o galpão que é a unidade experimental para o efeito do sistema de resfriamento de cobertura.

`nap` Número de aves perdidas (ou mortas) por dia.

conftemp é um `data.frame` com 456 observações e 6 variáveis, em que

`hora` As horas em cada dia, retomando do 0 em cada novo dia.

`hr` As horas a partir o primeiro dia continuamente.

`idade` A idade dos animais, em dias.

`h` Entalpia específica do ar.

`ctr` Carga térmica de radiação.

`itgu` Índice de temperatura de globo negro e umidade.

## Source

MACHADO, N. S.; TINÔCO, I. D. F. F.; ZOLNIER, S.; MOGAMI, C. A.; DAMASCENO, F. A.; ZEVIANI, W. M. Resfriamento da cobertura de aviários e seus efeitos na mortalidade e nos índices de conforto térmico. Nucleus, La Habana, v.9, n.2, 2012. <http://www.nucleus.feituverava.com.br/index.php/nucleus/article/view/718>, <http://www.ufv.br/dea/ambiagro/gallery/publicaço~oes/Artigo5.pdf>.

## Examples

```
#-----
# Gráfico da mortalidade das aves.

library(lattice)
library(latticeExtra)

str(confterm)
summary(confterm)

xtabs(~idade + resfr, data = confterm)

xyplot(nap ~ idade | resfr, data = confterm,
       groups = galp, type = "o",
       xlab = "Idade das aves (dias)",
       ylab = "Número de aves perdidas por galpão",
       strip = strip.custom(factor.levels = c(
           "Com sistema de resfriamento",
           "Sem sistema de resfriamento")),
       auto.key = list(corner = c(0.05, 0.9)))

#-----
# Gráfico das variáveis térmicas.

# Amplitude estendida das variáveis.
lim <- with(conftemp, apply(cbind(h, ctr, itgu), MARGIN = 2,
                          FUN = extendrange, f = 0.2))

# Anotação da eixo x do gráfico.
scales <- list(
  y = list(relation = "free"),
  x = list(at = seq(from = 1,
                    to = ceiling(max(conftemp$hr/24)) * 24,
                    by = 24)))
scales$x$labels <- seq_along(scales$x$at)

xyplot(h + ctr + itgu ~ hr, data = conftemp,
       outer = TRUE, type = "l", layout = c(1, NA),
       scales = scales, xlim = range(scales$x$at),
       xlab = "Dias",
       ylab = "Variáveis térmicas",
       panel = function(y, subscripts, ...) {
         wp <- which.packet()
```

```

r <- lim[, wp[1]]
panel.rect(10.5 + 24 * (scales$x$labels - 1), r[1],
           20 + 24 * (scales$x$labels - 1), r[2],
           col = "blue",
           border = "transparent",
           alpha = 0.25)
panel.xyplot(y = y, subscripts = subscripts, ...)
})

# Valores máximos do dia.
tempdia <- aggregate(cbind(hm = h, cm = ctr, im = itgu) ~ idade,
                    data = conftemp, FUN = max)

splom(tempdia[, -1])

confterm <- merge(confterm, tempdia, by = "idade")
str(confterm)

```

convergencez

*Avaliacao da Convergencia da Constante Normalizadora***Description**

Avalia a convergência da constante de normalização de um modelo COM-Poisson definida por:

$$Z = \sum \frac{\lambda^i}{(i!)^\nu}$$

, em que o parâmetro  $\nu$  é tomado como  $\exp \phi$ .

**Usage**

```
convergencez(model, tol = 1e-04, incremento = 10, maxit = 150,
            plot = TRUE)
```

**Arguments**

model	Objeto resultante da função <code>cmp</code> .
tol	Critério de parada do algoritmo, representa o valor tolerado para a diferença de $\frac{\lambda^i}{(i!)^\nu} - 0$ , pois no limite $i \rightarrow \infty$ o incremento $\frac{\lambda^i}{(i!)^\nu}$ tende a 0.
incremento	Número de incrementos da soma a serem considerados a cada iteração. Padrão definido como 10, ou seja, a cada iteração 10 incrementos são calculados.
maxit	Número máximo de iterações a serem realizadas pelo algoritmo. Se este número for atingido e o critério de tolerância não for atendido, uma mensagem de aviso será exibida.
plot	Argumento lógico. Se TRUE (padrão) os gráficos dos incrementos da constantes, calculadas para cada observação são exibidos.

**Value**

Uma lista com os incrementos das constantes  $Z$ ,  $Z(\lambda, \nu)$  da distribuição COM-Poisson, calculados para cada observação.

**Author(s)**

Eduardo E. R. Junior, <edujrrib@gmail.com>

**Examples**

```
m1 <- cmp(ncap ~ est * (des + I(des^2)), data = capdesfo)
tablez <- convergencez(m1)
str(tablez)

m2 <- cmp(ncap ~ dexp + I(dexp^2), data = capmosca)
tablez <- convergencez(m2)
str(tablez)
```

---

dcmp

---

*Probabilidades do Modelo Conway-Maxwell-Poisson*


---

**Description**

Calcula as probabilidades para uma variável aleatória distribuída conforme modelo COM-Poisson.

$$p(y, \lambda, \nu) = \frac{\lambda^y}{(y!)^\nu Z(\lambda, \nu)}$$

em que  $Z(\lambda, \nu)$  é a constante de normalização definida por  $\sum_{j=0}^{\infty} \frac{\lambda^j}{(j!)^\nu}$ . Nesta implementação o número de incrementos considerados para cálculo dessa constante é definido por `sumto`.  $\lambda > 0$  e  $\nu \geq 0$  são os parâmetros da distribuição.

**Usage**

```
dcmp(y, lambda, nu, sumto)
```

**Arguments**

<code>y</code>	Valor da variável de contagem.
<code>lambda</code>	Valor do parâmetro $\lambda$ da distribuição COM-Poisson.
<code>nu</code>	Valor do parâmetro $\nu$ da distribuição COM-Poisson.
<code>sumto</code>	Número de incrementos a serem considerados para a cálculo da constante normalizadora $Z$ .

**Author(s)**

Eduardo E. R. Junior, <edujrrib@gmail.com>

**Examples**

```
dpois(5, lambda = 5)
dcmp(5, lambda = 5, nu = 1, sumto = 20)

probs <- data.frame(y = 0:30)
within(probs, {
  py0 <- dpois(y, lambda = 15)
  py1 <- dcmp(y, lambda = 15, nu = 1, sumto = 50)
  py2 <- dcmp(y, lambda = 915, nu = 2.5, sumto = 50)
  py3 <- dcmp(y, lambda = 2.2, nu = 0.3, sumto = 50)
  plot(py0 ~ y, type = "h",
       ylim = c(0, max(c(py0, py2, py3))),
       ylab = expression(Pr(Y == y)))
  points(y + 0.1, py1, type = "h", col = 2)
  points(y - 0.3, py2, type = "h", col = 3)
  points(y + 0.3, py3, type = "h", col = 4)
  legend("topleft", bty = "n",
        col = c(1:4), lty = 1,
        legend = expression(
          Poisson(lambda == 15),
          CMP(lambda == 15, nu == 1),
          CMP(lambda == 915, nu == 2.5),
          CMP(lambda == 2.2, nu == 0.3)))
})
```

---

dgcnt

*Probabilidades do Modelo Gamma Count*

---

**Description**

Calcula as probabilidades para uma variável aleatória com distribuição Gamma Count:

$$p(y, \lambda, \alpha) = \left( \int_0^1 \frac{(\alpha\lambda)^{y\alpha}}{\Gamma(y\alpha)} u^{y\alpha-1} \exp\{-\alpha\lambda u\} du \right) - \left( \int_0^1 \frac{(\alpha\lambda)^{y\alpha}}{\Gamma((y+1)\alpha)} u^{(y+1)\alpha-1} \exp\{-\alpha\lambda u\} du \right),$$

em que  $\lambda > 0$  é a média da variável aleatória tempo entre eventos e  $\alpha > 0$  é o parâmetro de dispersão.

**Usage**

dgcnt(y, lambda, alpha)



**Arguments**

y	Valor da variável de contagem.
lambda	Valor do parâmetro $\lambda$ que é a média da distribuição do tempo entre eventos.
alpha	Valor do parâmetro $\alpha$ que é o parâmetro de dispersão.

**Value**

Retorna uma probabilidade, ou seja  $\Pr(Y = y|\lambda, \alpha) = p(y, \lambda, \alpha)$ .

**Author(s)**

Walmes Zeviani, <walmes@ufpr.br>.

**Examples**

```
dpois(5, lambda = 5)
dgcnt(5, lambda = 5, alpha = 1)

probs <- data.frame(y = 0:30)
within(probs, {
  py0 <- dpois(y, lambda = 15)
  py1 <- dgcnt(y, lambda = 15, alpha = 1)
  py2 <- dgcnt(y, lambda = 15, alpha = 0.5)
  py3 <- dgcnt(y, lambda = 15, alpha = 1.5)
  plot(py0 ~ y, type = "h",
        ylim = c(0, max(c(py0, py2, py3))),
        ylab = expression(Pr(Y == y)))
  points(y + 0.1, py1, type = "h", col = 2)
  points(y - 0.3, py2, type = "h", col = 3)
  points(y + 0.3, py3, type = "h", col = 4)
  legend("topleft", bty = "n",
        col = c(1:4), lty = 1,
        legend = expression(
          Poisson(lambda == 15),
          GC(lambda == 15, alpha == 1),
          GC(lambda == 15, alpha == 0.5),
          GC(lambda == 15, alpha == 1.5)))
})
```

**Description**

Calcula as probabilidades para uma variável aleatória com distribuição Poisson Generalizada na parametrização  $\lambda - \alpha$ :

$$p(y, \lambda, \alpha) = \left( \frac{\lambda}{1 + \alpha\lambda} \right)^y \frac{(1 + \alpha y)^{y-1}}{y!} \exp \left\{ -\lambda \left( \frac{1 + \alpha y}{1 + \alpha\lambda} \right) \right\},$$

em que  $\lambda > 0$  é a média da variável aleatória e  $\alpha$  é o parâmetro de dispersão, sendo que  $V(Y) = \lambda(1 + \alpha\lambda)^2$ . O espaço paramétrico de  $\alpha$  depende de  $\lambda$  e  $y$  pois  $1 + \alpha\lambda > 0$  e  $1 + \alpha y > 0$ .

**Usage**

```
dpgnz(y, lambda, alpha)
```

**Arguments**

y	Valor da variável de contagem.
lambda	Valor do parâmetro $\lambda$ que é a média da distribuição .
alpha	Valor do parâmetro $\alpha$ que é o parâmetro de dispersão.

**Value**

Retorna uma probabilidade, ou seja  $\Pr(Y = y|\lambda, \alpha) = p(y, \lambda, \alpha)$ .

**Author(s)**

Walmes Zeviani, <walmes@ufpr.br>

**Examples**

```
dpois(5, lambda = 5)
dpgnz(5, lambda = 5, alpha = 0)

probs <- data.frame(y = 0:30)
within(probs, {
  py0 <- dpois(y, lambda = 15)
  py1 <- dpgnz(y, lambda = 15, alpha = 0)
  py2 <- dpgnz(y, lambda = 15, alpha = 1/30)
  py3 <- dpgnz(y, lambda = 15, alpha = -1/30)
  plot(py0 ~ y, type = "h",
        ylim = c(0, max(c(py0, py2, py3))),
        ylab = expression(Pr(Y == y)))
  points(y + 0.1, py1, type = "h", col = 2)
  points(y - 0.3, py2, type = "h", col = 3)
  points(y + 0.3, py3, type = "h", col = 4)
  legend("topleft", bty = "n",
        col = c(1:4), lty = 1,
        legend = expression(
          Poisson(lambda == 15),
```

```
    PG(lambda == 15, alpha == 0),  
    PG(lambda == 15, alpha == 1/30),  
    PG(lambda == 15, alpha == -1/30)))  
  })
```

---

gcnt

*Ajuste do Modelo Gamma Count*

---

### Description

Estima os parâmetros de um modelo Gamma Count pela otimização da função de log-verossimilhança definida em `llgcnt`. A sintaxe assemelha-se com a função `glm` (Generalized Linear Models).

### Usage

```
gcnt(formula, data, start = NULL, ...)
```

### Arguments

<code>formula</code>	Um objeto da classe <code>formula</code> . Se necessária a inclusão de <i>offset</i> deve-se indicá-lo como <code>offset</code> .
<code>data</code>	Um objeto de classe <code>data.frame</code> que contém as variáveis descritas na <code>formula</code> .
<code>start</code>	Um vetor com os valores iniciais para os parâmetros do modelo necessários para o início do procedimento de estimação. Se <code>NULL</code> as estimativas de um modelo log-linear Poisson, com $\alpha = 0$ , são utilizadas como valores iniciais, pois uma chamada da <code>glm.fit</code> é feita internamente para obtê-los. O parâmetro $\alpha$ deve ser o primeiro elemento do vetor. Os restantes devem estar na posição correspondente às colunas da matriz gerada pelo argumento <code>formula</code> .
<code>...</code>	Argumentos opcionais do framework de maximização numérica <code>mle2</code> .

### Value

Um objeto de classe `mle2`, retornado da função de `mle2`, usada para estimação por máxima verossimilhança de modelos.

### Author(s)

Walmes Zeviani, <walmes@ufpr.br>.

### Examples

```
library(bbmle)  
  
str(soja)  
soja <- soja[-74, ]
```

```

m0 <- gcnt(nvag ~ bloc + umid * factor(K), data = soja)
m1 <- gcnt(nvag ~ bloc + umid + factor(K), data = soja)

anova(m0, m1)
summary(m1)

plot(profile(m1, which = "alpha"))
abline(v = 0, lty = 2, col = 2)

str(capdesfo)

n0 <- gcnt(ncap ~ est * (des + I(des^2)), data = capdesfo)
n1 <- gcnt(ncap ~ est + (des + I(des^2)), data = capdesfo)

anova(n0, n1)
summary(n0)

plot(profile(n0, which = "alpha"))

```

led

*Efeito das Cores de Iluminação de LED na Produção de Poedeiras***Description**

Experimento que estudou o efeito de 5 cores de iluminação de LED na produção de ovos de galinha poedeira. O experimento usou 360 aves divididas em 36 gaiolas e teve duração de 5 períodos de 14 dias cada. Foi empregado o delineamento quadrado latino de tamanho 5 períodos por 5 repartições. Em cada repartição haviam 6 gaiolas com 10 aves. Os ovos eram coletados diariamente de cada gaiola.

**Format**

Um `data.frame` com 2100 observações e 6 variáveis.

`period` Período de condução do experimento que teve uma duração de 5 períodos de 14 dias. A cada 14 dias, a iluminação era trocada de repartição. Os níveis desse fator são as linhas do quadrado latino.

`repart` Fator de níveis categóricos que representa as repartições do galpão. Cada repartição isolava 6 gaiolas para receberem apenas a iluminação de uma cor. Os níveis desse fator são as colunas do quadrado latino.

`corled` Fator de níveis categóricos cujos níveis são as cores emitida pela luz de LED: `amr` - amarelo, `azl` - azul, `brn` - branco, `vrd` - verde e `vrn` - vermelho. Esse fator é aplicado à todas as gaiolas de uma mesma repartição em um período.

`gaiola` A gaiola é a unidade amostral do experimento. Em cada gaiola haviam 10 aves e as variáveis resposta foram medidas nas gaiolas.

`dia` Número de dias após a troca da iluminação da repartição. A repartição permaneceu em cada cor de LED por 14 dias.

`ovos` Número de ovos produzidos por dia em cada gaiola.

**Source**

Dados referentes ao Capítulo 2 da Dissertação de Mestrado de Rodrigo Borille.

Rodrigo, B. (2013). LED de diferentes cores como alternativa sustentável para a iluminação de poedeiras comerciais. Dissertação (Mestrado em Zootecnia). Universidade Federal da Grande Dourados, Dourados: MS. <http://200.129.209.183/arquivos/arquivos/78/MESTRADO-ZOOTECNIA/Dissertaç~aoRodrigoBorille.pdf>

**Examples**

```
library(lattice)
library(latticeExtra)

data(led)
str(led)

ftable(xtabs(~period + corled + repart, data = led))

useOuterStrips(
  xyplot(ovos ~ dia | repart + period,
        data = led, groups = corled,
        col = c("yellow", "blue", "gray30", "green", "red"),
        type = c("p", "smooth"), jitter.x = TRUE, as.table = TRUE,
        xlab = "Dias",
        ylab = "Número de ovos produzidos"))

useOuterStrips(histogram(~ovos | repart + period, data = led,
                        groups = corled, as.table = TRUE,
                        breaks = seq(min(led$ovos) - 0.5,
                                    max(led$ovos) + 0.5, by = 1),
                        xlab = "Número de ovos produzidos",
                        ylab = "Frequência relativa (%)"))

xyplot(ovos ~ corled | period, data = led,
       type = c("p", "a"), jitter.x = TRUE,
       ylab = "Número de ovos produzidos",
       xlab = "Cor emitida pelo LED")
```

---

llcmp

NA

---

**Description**

Calcula a log-verossimilhança de um modelo de regressão para o parâmetro  $\lambda$  considerando as respostas de contagem ( $y$ ), condicionadas as suas covariáveis ( $X$ ), distribuídas conforme modelo COM-Poisson.

**Usage**

```
llcmp(params, y, X, sumto = ceiling(max(y)^1.2))
```

**Arguments**

params	Um vetor de parâmetros do modelo COM-Poisson. O primeiro elemento desse vetor deve ser o parâmetro de dispersão do modelo, $\phi$ , os restantes são os parâmetros $\beta$ 's associados ao preditor linear em $\lambda$ .
y	Um vetor com variável dependente do modelo, resposta do tipo contagem.
X	A matriz de delineamento correspondente ao modelo linear ligado à $\lambda$ pela função de ligação log. A matriz do modelo pode ser construída com a função <code>model.matrix</code> .
sumto	Número de incrementos a serem considerados para a cálculo das constantes normalizadoras. Como padrão, o número de incrementos é o valor inteiro de $(\max y)^{1.5}$ , porém esse valor padrão não é ótimo. Uma avaliação da escolha desse argumento, pós ajuste pode ser realizada via <code>convergencez</code> .

**Details**

A função de log-verossimilhança da COM-Poisson, na parametrização de modelo de regressão é:

$$\ell(\beta, \nu, y) = \sum_{i=1}^n y_i \log(\lambda_i) - \nu \sum_{i=1}^n \log(y!) - \sum_{i=1}^n \log(Z(\lambda_i, \nu))$$

em que (i)  $\lambda_i = \exp(X_i\beta)$ , no modelo de regressão COM-Poisson um preditor linear é ligado à  $\lambda$  por meio da função de ligação log. Note que não estamos modelando diretamente a média, assim as estimativas dos parâmetros  $\beta$  não tem a mesma interpretação dos modelos Poisson, por exemplo. Contudo, os sinais desses parâmetros indicam efeitos de acréscimo ou decréscimo nas contagens médias. (ii)  $\nu$  é o parâmetro de dispersão que indica equi, sub ou superdispersão das contagens  $y$ . Sendo  $\nu = 1$  o caso de equidispersão,  $0 \leq \nu < 1$  superdispersão e  $\nu > 1$  subdispersão. Vale ressaltar que a variância  $V(Y)$  não tem expressão fechada e não é definida unicamente por  $\nu$ . (iii)  $Z(\lambda_i, \nu)$  é a constante de normalização definida por

$$\sum_{j=0}^{\infty} \frac{\lambda_i^j}{(j!)^\nu}$$

. Note que são calculadas  $n$  constantes  $Z$ . Nesta implementação o número de incrementos considerados para cálculo dessas constantes é definido por `sumto`, o mesmo número de incrementos é considerado para o cálculo de todas as constantes. Uma verificação pós ajuste da escolha de `sumto` pode ser realizada a partir de `convergencez`.

Nesta parametrização o modelo COM-Poisson tem como casos particulares os modelos Poisson quando  $\nu = 1$ , Bernoulli quando  $\nu \rightarrow \infty$  (ou o modelo logístico considerando modelos de regressão) e Geométrico quando  $\nu = 0$  e  $\lambda < 1$ .

Para que não seja necessário restringir o algoritmo de maximização da log-verossimilhança, a função foi implementada reparametrizando o parâmetro  $\nu$  para  $\log(\phi)$ . Assim o parâmetro estimado será  $\phi$  que tem suporte nos reais, assim como o vetor  $\beta$ .

**Value**

O negativo da log-verossimilhança do modelo Conway-Maxwell-Poisson com os parâmetros e dados informados.

**Author(s)**

Eduardo E. R. Junior, <edujrrib@gmail.com>

**See Also**

[mle2](#)

---

 llgcnt

---

*Log-Verossimilhança do Modelo Gamma Count*


---

**Description**

Calcula a log-verossimilhança de um modelo de regressão Gamma Count para a resposta de contagem ( $y$ ).

**Usage**

```
llgcnt(params, y, X, offset = NULL)
```

**Arguments**

params	Um vetor de (estimativas dos) parâmetros do modelo de regressão. O primeiro elemento desse vetor é o parâmetro de dispersão do modelo e os restantes são parâmetros de locação. Essa função retorna o negativo da log-verossimilhança pois foi construída para ser usada na <a href="#">mle2</a> .
y	Um vetor com variável dependente do modelo, resposta do tipo contagem.
X	A matriz de delineamento correspondente ao modelo linear ligado à média pela função de ligação log. A matriz do modelo pode ser construída com a função <a href="#">model.matrix</a> .
offset	Um vetor, de mesmo comprimento de $y$ , com valores que correspondem aos offsets (ou exposição) para cada valor observado. Se NULL, é usado 1 como offset.

**Details**

A função de verossimilhança para uma observação,  $L$ , da Gamma Count é:

$$L(\lambda, \alpha; y) = \left( \int_0^T \frac{(\alpha\lambda)^{y\alpha}}{\Gamma(y\alpha)} u^{y\alpha-1} \exp\{-\alpha\lambda u\} du \right) - \left( \int_0^T \frac{(\alpha\lambda)^{y\alpha}}{\Gamma((y+1)\alpha)} u^{(y+1)\alpha-1} \exp\{-\alpha\lambda u\} du \right).$$

Se  $\tau \sim \text{Gamma}(\alpha, \alpha\lambda)$ , então  $E(\tau) = \frac{\alpha}{\alpha\lambda} = \frac{1}{\lambda}$  e  $V(\tau) = \frac{1}{\alpha\lambda^2}$ .

Usando  $G(\cdot)$  para representar o resultado de cada uma das integrais entre parenteses, que corresponde a probabilidade acumulada de uma variável aleatória gamma com parâmetros  $y\alpha$  e  $\alpha\lambda$ , tem-se

$$L(\lambda, \alpha; y) = G(T, y\alpha, \alpha\lambda) - G(T, (y + 1)\alpha, \alpha\lambda).$$

A função log-verossimilhança de uma observação é, portanto,

$$\ell(\lambda, \alpha; y) = \ln [G(T, y\alpha, \alpha\lambda) - G(T, (y + 1)\alpha, \alpha\lambda)].$$

Para uma amostra aleatória independente, a função de log-verossimilhança é

$$\ell(\lambda, \alpha; y) = \sum_{i=1}^n \ln [G(T, y\alpha, \alpha\lambda) - G(T, (y + 1)\alpha, \alpha\lambda)].$$

Nestas expressões,  $\alpha$  é o parâmetro de dispersão da variável aleatória  $Y$  sendo que se  $\alpha = 1$  então  $Y \sim \text{Poisson}$ , se  $\alpha < 1$  então  $V(Y) > E(Y)$  e  $\alpha > 1$  então  $V(Y) < E(Y)$ .

Como  $\alpha$  e  $\lambda$  devem ser positivos, usou-se a função de ligação log para escrever a log-verossimilhança com domínio nos reais para os parâmetros.

### Value

O negativo da log-verossimilhança do modelo Gamma Count para os parâmetros e dados informados.

### Author(s)

Walmes Zeviani, <walmes@ufpr.br>.

### See Also

[mle2](#).

### Examples

```
set.seed(123)
y <- rpois(10, lambda = 5)

# Log-verossimilhança pela Poisson.
sum(dpois(y, lambda = 5, log = TRUE))

# Log-verossimilhança pela GCNT usando alpha = 0
llgcnt(params = c(0, log(5)), y = y, X = cbind(y * 0 + 1))

set.seed(121)
y <- rpois(100, lambda = exp(1))
X <- cbind(0 * y + 1)

grid <- expand.grid(alpha = seq(-0.5, 0.5, by = 0.02),
```



```

                                lambda = seq(0.1, 2.1, by = 0.05))
str(grid)

grid$ll <- apply(grid, MARGIN = 1,
                FUN = function(vec) {
                    llgcnt(params = vec, y = y, X = X,
                          offset = NULL)
                })

library(latticeExtra)

levelplot(ll ~ alpha + lambda, data = grid) +
  layer(panel.abline(v = 0, h = 1, lty = 2))

```

llpgnz

*Log-Verossimilhanca do Modelo Poisson Generalizada***Description**

Calcula a log-verossimilhança de um modelo de regressão para a média com distribuição Poisson Generalizada para a resposta de contagem ( $y$ ).

**Usage**

```
llpgnz(params, y, X, offset = NULL)
```

**Arguments**

params	Um vetor de (estimativas dos) parâmetros do modelo de regressão. O primeiro elemento desse vetor é o parâmetro de dispersão do modelo e os restantes são parâmetros de locação. Essa função retorna o negativo da log-verossimilhança pois foi construída para ser usada na <a href="#">mle2</a> .
y	Um vetor com variável dependente do modelo, resposta do tipo contagem.
X	A matriz de delineamento correspondente ao modelo linear ligado à média pela função de ligação log. A matriz do modelo pode ser construída com a função <a href="#">model.matrix</a> .
offset	Um vetor, de mesmo comprimento de $y$ , com valores que correspondem aos offsets (ou exposição) para cada valor observado. Se NULL, é usado 1 como offset.

**Details**

A função de log-verossimilhança da Poisson Generalizada, na parametrização de modelo de regressão é:

$$\ell(\lambda, \alpha, y) = y(\log(\lambda) - \log(1 + \alpha\lambda)) + (y - 1) \log(1 + \alpha y) - \lambda \left( \frac{1 + \alpha y}{1 + \alpha \lambda} \right) - \log(y!),$$

em que  $\alpha$  é o parâmetro de dispersão e  $\lambda > 0$  é a média  $E(Y) = \lambda$  e  $y = 0, 1, \dots$  é vetor observado da variável de contagem. Nessa parametrização,  $V(Y) = \lambda(1 + \alpha\lambda)^2$ . A Poisson Generalizada em a Poisson como caso particular quando  $\alpha = 0$ .

Para o modelo de regressão, um preditor linear é ligado à média pela função de ligação log,  $\log(\lambda) = X\beta$ , tal como é para o modelo Poisson com link log.

O espaço paramétrico de  $\alpha$  não limitado para o lado direito do zero ( $\alpha$  positivo) mas para o lado esquerdo ( $\alpha$  negativo) o limite inferior é dependente do parâmetro  $\lambda$  e dos valores observados de  $y$ . Valores não finitos podem ser retornados durante a estimação quando  $1 + \alpha\lambda$  ou  $1 + \alpha y$  não forem maiores que zero.

### Value

O negativo da log-verossimilhança do modelo Poisson Generalizado para os parâmetros e dados informados.

### Author(s)

Walmes Zeviani, <walmes@ufpr.br>

### See Also

[mle2](#).

### Examples

```
set.seed(123)
y <- rpois(10, lambda = 5)

# Log-verossimilhança pela Poisson.
sum(dpois(y, lambda = 5, log = TRUE))

# Log-verossimilhança pela PGNZ usando alpha = 0
llpgnz(params = c(0, log(5)), y = y, X = cbind(y * 0 + 1))

set.seed(121)
y <- rpois(100, lambda = exp(1))
X <- cbind(0 * y + 1)

grid <- expand.grid(alpha = seq(-0.1, 0.4, by = 0.01),
                  lambda = seq(0.1, 2.1, by = 0.025))

grid$ll <- apply(grid, MARGIN = 1,
                FUN = function(vec) {
                  llpgnz(params = vec, y = y, X = X, offset = NULL)
                })

library(latticeExtra)

levelplot(ll ~ alpha + lambda, data = grid) +
  layer(panel.abline(v = 0, h = 1, lty = 2))
```

**Description**

Esse é um pacote com o material do Curso "Modelos de Regressão para Dados de Contagem" oferecido pelos autores na 61 RBRAS ([rbras2016.ufba.br](http://rbras2016.ufba.br)) que aconteceu entre 23 e 25 de Maio em Salvador, Bahia.

O objetivo desse Curso é apresentar diferentes modelos de regressão para análise de dados de contagem, discutir suas principais propriedades e ilustrar suas aplicações em dados reais por meio do software R. Dentre os modelos considerados estão o modelo Poisson, algumas extensões para acomodação de super (ou sub) dispersão (Quasi-Poisson, COM-Poisson, Count-Gamma, Binomial-Nagativo), modelos para dados inflacionados de zeros e com a inclusão de efeitos aleatórios.

Neste pacote estão disponíveis as vignettes que mostram como coordenar funções do R para análise dos dados, os slides do Curso, as funções e alguns dos dados utilizados nos estudos de caso.

**Description**

Resultados de um experimento em casa de vegetação que estudou a reprodução de nematóides em cultivares/linhagens de feijoeiro. O solo dos vasos foi inicialmente contaminado com namatóides e as parcelas tiveram duas plantas. Ao final do experimento, as raízes das duas plantas por parcela foram lavadas, trituradas, peneiradas e diluídas para fazer a contagem dos nematóides em alíquotas dessa solução.

**Format**

Um `data.frame` com 94 observações e 4 variáveis.

`cult` Fator categórico que indica a linhagem de feijoeiro semeada em vasos com solo contaminado com nematóide.

`mfr` Massa fresca de raízes (g) produzida por parcela (duas plantas) que foi lavada, triturada, peneirada e diluída para fazer a contagem dos nematóides.

`vol` Volume (ml) usado para diluir a massa fresca de raízes. Esse volume foi agitado para homogeneização e depois uma alíquota de 1 ml foi extraída e colocada em uma lâmina de contagem.

`nema` Número de nematóides na alíquota de 1 ml, determinado por contagem direta na lâmina.

`off` É o offset da contagem, o equivalente em massa de fresca de raízes de uma alíquota de 1 ml, ou seja,  $off = mfr/vol$ .

### Source

Cedido para fins acadêmicos por Andressa Cristina Zamboni Machado, pesquisadora do Instituto Agrônômico do Paraná (IAPAR), e pelo técnico agrícola do IAPAR Santino Aleandro da Silva.

O nome das cultivares, a espécie do nematóide e outras informações não foram dadas para preservar a originalidade da Pesquisa.

### Examples

```
m0 <- glm(nema ~ offset(log(off)) + cult,
          data = nematoide,
          family = poisson)

# Diagnóstico.
par(mfrow = c(2, 2))
plot(m0); layout(1)

# Estimativas dos parâmetros.
summary(m0)

# Quadro de deviance.
anova(m0, test = "Chisq")

library(bbmle)

# Poisson Generalizada.
m1 <- pgnz(formula(m0), data = nematoide)

# Diferença de deviance.
2 * diff(c(logLik(m0), logLik(m1)))

# Estimativas dos parâmetros.
summary(m1)
```

---

ninfas

*Ninfas de Mosca-Branca em Lavoura de Soja*

---

### Description

Experimento conduzido em casa de vegetação sob o delineamento de blocos casualizados. No experimento foram avaliadas plantas de diferentes cultivares de soja contabilizando o número de ninfas de mosca-branca nos folíolos dos terços superior, médio e inferior das plantas. As avaliações ocorreram em 6 datas dentre os 38 dias do estudo.

## Format

Um `data.frame` com 240 observações e 8 variáveis.

`data` Data em que foram avaliadas as plantas de soja.

`dias` Inteiro que indica o número de dias após o experimento no ato da avaliação.

`cult` Fator com a identificação da cultivar de soja. Foram 10 cultivares avaliadas neste experimento.

`bloco` Fator com 4 níveis que representam os blocos utilizados para controle de variação local.

`nsup` Número de ninfas de mosca-branca nos folíolos do terço superior.

`nmed` Número de ninfas de mosca-branca nos folíolos do terço médio.

`ninf` Número de ninfas de mosca-branca nos folíolos do terço inferior.

`ntot` Número de ninfas de mosca-branca considerando todos os folíolos (soma de `nsup`, `nmed`, `ntot`).

## References

Suekane, R., Degrande, P. E., de Lima Junior, I. S., de Queiroz, M. V. B. M., & Rigoni, E. R. (2013). Danos da Mosca-Branca *Bemisia Tabaci* e distribuição vertical das ninfas em cultivares de soja em casa de vegetação. Arquivos do Instituto Biológico, 80(2), 151-158. [Dissertação de Renato Suekane.](#)

## Examples

```
data(ninfas)
str(ninfas)

xtabs(~data + cult, data = ninfas)

library(lattice)

xyplot(ntot ~ dias | cult,
       data = ninfas,
       type = c("p", "spline"),
       grid = TRUE,
       as.table = TRUE,
       layout = c(NA, 2))

# Somente as cultivares que contém BRS na identificação
da <- droplevels(subset(ninfas, grepl("BRS", x = cult)))

xyplot(ntot ~ dias | cult,
       data = da,
       type = c("p", "spline"),
       grid = TRUE,
       as.table = TRUE,
       layout = c(NA, 2))
```

---

`panel.beeswarm`*Diagrama de Dispersao com Aranjo dos Pontos como Colmeia*

---

### Description

Used to make scatter plot of discrete variables with no overlapping points. Observations with the same y value are spread.

### Usage

```
panel.beeswarm(x, y, subscripts, spread, ...)
```

### Arguments

`x`, `y`, `subscripts`, ...

Argumentos passados para a [panel.xyplot](#).

`spread`

Um escalar numérico a distância entre os pontos. Esse valor é obtido por tentativa erro e toda vez que mudar as dimensões do gráfico, eles precisam ser novamente fornecidos, no entanto são valores na escala do eixo x e por isso são baseados nas distâncias entre os níveis do fator representado neste eixo. Como sugestão, abra sempre a janela gráfica (`x11()`) ou faça a exportação (`png()`, `pdf()`, etc) com dimensões conhecidas e calibre o `spread` para que seja exibido adequadamente.

### Value

A função passa conteúdo para o argumento `panel`.

### Author(s)

Walmes Zeviani baseado no pacote [beeswarm](#).

### See Also

[xyplot](#).

### Examples

```
data(capdesfo)
str(capdesfo)

library(lattice)

# x11(width = 7, height = 2.8)
xyplot(ncap ~ des | est, data = capdesfo,
       layout = c(5, 1), as.table = TRUE,
       type = c("p", "smooth"), col = 1, col.line = "gray50",
```

```

xlim = extendrange(c(0:1), f = 0.15),
xlab = "Nível de desfolha artificial",
ylab = "Número de capulho produzidos",
spread = 0.07, panel = panel.beeswarm)

# x11(width = 7, height = 2.8)
xyplot(ncap ~ est | factor(des), data = capdesfo,
       layout = c(5, 1), as.table = TRUE,
       type = c("p", "smooth"), col = 1, col.line = "gray50",
       xlab = "Fase de desenvolvimento da planta",
       ylab = "Número de capulhos produzidos",
       scales = list(x = list(
         at = 1:nlevels(capdesfo$est),
         labels = substr(levels(capdesfo$est),
           start = 1, stop = 3))),
       spread = 0.35, panel = panel.beeswarm)

```

panel.cbH

*Funcao para Intervalos e Bandas de Confianca com a Lattice***Description**

Essa função permite representar intervalos de confiança e bandas de confiança em gráficos do pacote lattice.

**Usage**

```

panel.cbH(x, y, ly, uy, subscripts, cty, col.line = plot.line$col,
         lwd = plot.line$lwd, desloc = NULL, fill = 1, alpha = 0.1,
         length = 0.05, ...)

prepanel.cbH(y, ly, uy, subscripts)

```

**Arguments**

`x`, `y`, `subscripts`, `col.line`, `lwd`, ...  
Argumentos passados para a [panel.xyplot](#) pela função [xyplot](#).

`ly` Limite inferior do intervalo/banda de confiança.

`uy` Limite superior do intervalo/banda de confiança.

`cty` Uma string que indica o tipo de representação. Atualmente são aceitos os valores "bars" para intervalos de confiança e "bands" para bandas de confiança.

`desloc` Um vetor numérico nos quais os valores representam quantidades a somar/subtrair dos valores de `x` para não sobrepor intervalos. Com esse argumento pode-se representar mais de um intervalo por valor de `x`. Não é usado quando `cty = "bands"`.

fill	Uma representação de cor para preencher o interior das bandas de confiança. Não é usado quando <code>cty = "bars"</code> .
alpha	Nível de transparência na cor de preenchimento das bandas de confiança para permitir sobrepor bandas. O valor default é 0.1. Não é usado quando <code>cty = "bars"</code> .
length	Comprimento para as extremidades dos intervalos que formam o "T". O valor default é 0.05. Não é usado quando <code>cty = "bands"</code> .

**Value**

São usadas dentro de funções do pacote **lattice**.

**Author(s)**

Walmes Zeviani baseado na lista de discussão R-help.

**See Also**

[prepanel.cbH](#).

**Examples**

```
library(lattice)
library(latticeExtra)

m0 <- lm(sqrt(dist) ~ speed, data = cars)
pred <- with(cars, data.frame(speed = seq(min(speed),
                                       max(speed),
                                       length.out = 20)))
aux <- predict(m0, newdata = pred, interval = "confidence")
pred <- cbind(pred, aux)

xyplot(sqrt(dist) ~ speed, data = cars) +
  as.layer(xyplot(fit ~ speed, data = pred, type = "l",
                ly = pred$lwr, uy = pred$upr, cty = "bands",
                fill = "blue", alpha = 0.3,
                prepanel = prepanel.cbH,
                panel = panel.cbH))

m1 <- lm(weight ~ feed, data = chickwts)
pred <- with(chickwts, data.frame(feed = levels(feed)))
aux <- predict(m1, newdata = pred, interval = "confidence")
pred <- cbind(pred, aux)

xyplot(weight~feed, data=chickwts)+
  as.layer(xyplot(fit ~ feed, data = pred,
                ly = pred$lwr, uy = pred$upr,
                cty = "bars",
                prepanel = prepanel.cbH,
                desloc = rep(0.15, length(pred$fit)),
                panel = panel.cbH))
```



```

da <- expand.grid(trt = gl(2, 1), x = 1:7)
da$y <- with(da, as.integer(trt) + 0.5 * x + rnorm(x, 0, 0.4))
xyplot(y ~ x, groups = trt, data = da)

m2 <- lm(y ~ trt + x, data = da)

pred <- with(da, expand.grid(trt = levels(trt),
                           x = seq(min(x), max(x), length.out = 20)))
aux <- predict(m2, newdata = pred, interval = "confidence")
pred <- cbind(pred, aux)

xyplot(y ~ x, groups = trt, data = da) +
  as.layer(xyplot(fit ~ x, groups = trt, data = pred, type = "l",
                 ly = pred$lwr, uy = pred$upr,
                 cty = "bands", alpha = 0.25,
                 prepanel = prepanel.cbH,
                 panel = panel.superpose,
                 panel.groups = panel.cbH))

```

---

panel.groups.segplot *Painel para Fazer Intervalos para Grupos no segplot()*

---

### Description

Essa função permite fazer intervalos de confiança (ou com barras de erro) em gráficos da `latticeExtra::segplot()` para grupos (`groups =`) de tal forma que eles não fiquem sobrepostos.

### Usage

```
panel.groups.segplot(x, y, z, centers, groups, gap = NULL, data, subscripts,
  ...)
```

### Arguments

`x`, `y`, `z`, `centers`, `data`, `subscripts`, ...  
Argumentos passados para a `segplot`.

`groups` A variável (factor) de agrupamento, de mesmo comprimento de `lwr` e `upr`.

`gap` Escalar que representa a distância entre os segmentos. O valor default é 0,1. Como um fator com  $k$  níveis é representado pelos números inteiros  $1, 2, \dots, k$ , então  $0 \leq \text{gap} < 1/k$ . Se for usado um valor negativo, os intervalos serão apresentados em ordem inversa.

### Value

A função é passada para o argumento `panel` e retorna elementos necessários para a `segplot`.

**Author(s)**

Walmes Zeviani

**See Also**

[segplot](#).

**Examples**

```
library(latticeExtra)

m0 <- lm(log(breaks) ~ wool * tension, data = warpbreaks)

pred <- with(warpbreaks, expand.grid(KEEP.OUT.ATTRS = TRUE,
                                   wool = levels(wool),
                                   tension = levels(tension)))

pred <- cbind(pred,
              predict(m0, newdata = pred, interval = "confidence"))
str(pred)

segplot(wool ~ lwr + upr, centers = fit, data = pred,
        draw = FALSE, horizontal = FALSE)

segplot(wool ~ lwr + upr, centers = fit, data = pred,
        draw = FALSE, horizontal = FALSE,
        groups = tension, gap = NULL,
        panel = panel.groups.segplot)
```

---

peixe

*Peixes Capturados por Pescadores em um Parque Estadual*

---

**Description**

Dados sobre 250 grupos que foram ao parque ao parque estadual para pescar. As informações coletadas foram referentes a presença ou não de um campista, ao número de crianças no grupo e ao número de indivíduos no grupo.

**Format**

Um `data.frame` com 250 observações e 4 variáveis.

`campista` Fator com dois níveis que representa a presença (1) ou ausência (0) de um campista no grupo.

`ncrianças` Número de crianças no grupo.

`npessoas` Número total de pessoas no grupo.

`npeixes` Número de peixes capturados pelo grupo.

## References

Calvin, J. A. (1998). Regression Models for Categorical and Limited Dependent Variables. *Technometrics*, 40(1), 80-81.

## Examples

```
data(peixe)
(proptb <- prop.table(table(peixe$npeixes)))
plot(proptb)

library(lattice)
# Contagens (marginal aos efeitos das covariáveis)
histogram(~npeixes, data = peixe, nint = 50)

# Contagens com relação as covariáveis
xyplot(npeixes ~ ncriancas + npessoas,
       groups = campista,
       data = peixe,
       jitter.x = TRUE,
       type = c("p", "g", "smooth"))
```

---

pgnz

*Ajuste do Modelo Poisson Generalizado*

---

## Description

Estima os parâmetros de um modelo Poisson Generalizado pela otimização da função de log-verossimilhança definida em `llpgnz`. A sintaxe assemelha-se com a função `glm` (Generalized Linear Models).

## Usage

```
pgnz(formula, data, start = NULL, ...)
```

## Arguments

<code>formula</code>	Um objeto da classe <code>formula</code> . Se necessária a inclusão de <i>offset</i> deve-se indicá-lo como <code>offset</code> .
<code>data</code>	Um objeto de classe <code>data.frame</code> que contém as variáveis descritas na <code>formula</code> .
<code>start</code>	Um vetor com os valores iniciais para os parâmetros do modelo necessários para o início do procedimento de estimação. Se <code>NULL</code> as estimativas de um modelo log-linear Poisson, com $\alpha = 0$ , são utilizadas como valores iniciais, pois uma chamada da <code>glm.fit</code> é feita internamente para obtê-los. O parâmetro $\alpha$ deve ser o primeiro elemento do vetor. Os restantes devem estar na correspondente às colunas da matriz gerada pelo argumento <code>formula</code> .
<code>...</code>	Argumentos opcionais do framework de maximização numérica <code>mle2</code> .

**Value**

Um objeto de classe `mle2`, retornado da função de `mle2`, usada para estimação por máxima verossimilhança de modelos.

**Author(s)**

Walmes Zeviani, <walmes@ufpr.br>

**Examples**

```
library(bbmle)

str(soja)
soja <- soja[-74, ]

m0 <- pgz(nvag ~ bloc + umid * factor(K), data = soja)
m1 <- pgz(nvag ~ bloc + umid + factor(K), data = soja)

anova(m0, m1)
summary(m1)

plot(profile(m1, which = "alpha"))
abline(v = 0, lty = 2, col = 2)

str(capdesfo)

n0 <- pgz(ncap ~ est * (des + I(des^2)), data = capdesfo)
n1 <- pgz(ncap ~ est + (des + I(des^2)), data = capdesfo)

anova(n0, n1)
summary(n0)

plot(profile(n0, which = "alpha"))
```

---

postura

*Numero de Trocas de Postura em Ovelhas*

---

**Description**

Os dados são referentes a um experimento realizado com o objetivo de investigar o efeito de uma intervenção, por parte do cuidador, no comportamento de ovelhas.

Para isso, foram consideradas ovelhas de duas linhagens distintas (pouco ou muito reativas), submetidas a dois tipos diferentes de intervenção (observação ou observação + intervenção).

A variável resposta considerada é o número de mudanças na postura corporal do animal ao longo do período de observação (3 minutos).

**Format**

Um `data.frame` com 38 observações e 3 variáveis, em que

- `trat` Fator com dois níveis que representa a intervenção feita.
- `linh` Fator que representa a linhagem das ovelhas.
- `npost` Número de trocas de postura em 3 minutos.

**Examples**

```
tb <- xtabs(~trat + linh, data = postura)
tb

mosaicplot(tb)
```

---

seguros	NA
---------	----

---

**Description**

Dados referentes ao acompanhamento de clientes de uma seguradora de automóveis ao longo de um ano. Foram registrados as variáveis descritas abaixo para 16483 clientes.

**Format**

Um `data.frame` com 16483 observações e 7 variáveis.

- `tipo` Tipo de veículo segurado. Fator com seis níveis hatch, mono, picape, sedan, wagon e suv.
- `idade` Idade do cliente, em anos.
- `sexo` Sexo do cliente.
- `civil` Estado civil do cliente. Fator com dois níveis, M para clientes do sexo masculino e F para feminino.
- `valor` Valor do veículo segurado, em 1000 reais.
- `expos` Período de cobertura do cliente durante o ano sob análise.
- `nsinist` Número de sinistros registrados no período.

**Examples**

```
data(seguros)

str(seguros)

summary(seguros)
```

---

soja

*Umidade do Solo e Doses de Potássio na Cultura da Soja*

---

### Description

Variáveis de contagem de um experimento fatorial  $5 \times 3$  que estudou níveis de adubação com potássio e níveis de umidade do solo nos componentes de produção da soja. O experimento foi instalado em casa de vegetação no delineamento de blocos casualizados completos e a unidade experimental foi um vaso com duas plantas de soja. No experimento foram medidas várias respostas, sendo que o número de vagens por vaso e o número de grãos por vaso foram as variáveis de contagem.

### Format

Um `data.frame` com 75 observações e 5 variáveis, em que

`K` Fator de níveis métricos. Dose de potássio aplicado ao solo na forma de adubação, em  $\text{mg dm}^{-3}$ .

`umid` Fator de níveis categóricos. Faixas de umidade do solo. Procurou-se por meio de pesagens diárias dos vasos, manter a umidade do solo nas faixas de 35 a 40; 47,5 a 52,5; e 60 a 65 do volume total de poros.

`bloc` Fator categórico para controle local no experimento. Foram controladas as variações de ambiente da casa de vegetação, como exposição ao sol, e mão de obra ao longo do experimento, como irrigação e capina dos vasos e controle de doenças.

`ngra` Número de grãos por vaso na colheita da soja.

`nvag` Número de vagens viáveis por vaso na colheita da soja.

### Details

A observação número 74 (bloco V, K 120, `umid` 62,5) foi uma parcela onde as plantas não tiveram um desenvolvimento normal. Portanto, pode-se remover esta observação para análise dos dados.

### Source

Serafim, Milson Evaldo; Ono, Fábio Benedito; Zeviani, Walmes Marques; Novelino, José Oscar; Silva, Joil Vilhalva. (2012). Umidade do solo e doses de potássio na cultura da soja. *Revista Ciência Agronômica*, 43(2), 222-227. <https://dx.doi.org/10.1590/S1806-66902012000200003>, <http://www.scielo.br/pdf/rca/v43n2/a03v43n2.pdf>

### Examples

```
library(lattice)

data(soja)
str(soja)

xtabs(~umid + K, data = soja[-75, ])
```

```
xyplot(nvag + ngra ~ K, groups = umid, outer = TRUE,
       data = soja[-74, ],
       type = c("p", "a"), scales = "free",
       ylab = NULL,
       xlab = expression("Nível de potássio aplicado"~(mg~dm^{-3})),
       auto.key = list(title = "Umidade do solo (%)",
                       cex.title = 1,
                       columns = 3),
       strip = strip.custom(
         factor.levels = c("Número de vagens",
                           "Número de grãos")))
```

# Index

\*Topic **excesso-zeros**  
seguros, [37](#)

apc, [3](#)

calc\_mean\_cmp, [4](#)

calc\_mean\_gcnt, [5](#)

calc\_var\_cmp, [6](#)

cambras, [6](#)

capdesfo, [8](#)

capmosca, [9](#)

cmp, [11](#), [14](#)

conftemp (confterm), [12](#)

confterm, [12](#)

convergencez, [11](#), [14](#), [22](#)

dcmp, [15](#)

dgcnt, [16](#)

dpgnz, [17](#)

formula, [11](#), [19](#), [35](#)

gcnt, [19](#)

glm, [11](#), [19](#), [35](#)

glm.fit, [11](#), [19](#), [35](#)

led, [20](#)

llcmp, [11](#), [21](#)

llgcnt, [19](#), [23](#)

llpgnz, [25](#), [35](#)

LSmatrix, [3](#)

mle2, [11](#), [19](#), [23–26](#), [35](#), [36](#)

model.matrix, [22](#), [23](#), [25](#)

MRDCr, [27](#)

MRDCr-package (MRDCr), [27](#)

nematoide, [27](#)

ninfas, [28](#)

offset, [19](#), [35](#)

panel.beeswarm, [30](#)

panel.cbH, [31](#)

panel.groups.segplot, [33](#)

panel.xyplot, [30](#), [31](#)

peixe, [34](#)

pgnz, [35](#)

postura, [36](#)

prepanel.cbH, [32](#)

prepanel.cbH (panel.cbH), [31](#)

segplot, [33](#), [34](#)

seguros, [37](#)

soja, [38](#)

xyplot, [30](#), [31](#)