Modelo linear generalizado

Distribuição Poisson

2023

Os dados deste exemplo, apresentados abaixo, referem-se a contagens de partículas de vírus em um líquido para cinco diluições diferentes (volumes diferentes). O objetivo consiste em estimar o número médio de partículas de vírus por unidade de volume (uma taxa), verificando se este número varia com a diluição.

O parâmetro de interesse (número médio de partículas de vírus por unidade de volume) é denotado por λ, de modo que a média da variável resposta (contagem de partículas de vírus), denotada por μ, é dada por

$$\mu = \lambda x, \tag{1}$$

em que x representa a variável explicativa (diluição) e λ pode variar com a diluição.

Foram utilizadas cinco diluições diferentes. Foram realizadas quatro repetições para as quatro primeiras diluições e cinco repetições para a última diluição de forma que o tamanho da amostra é 21. Os dados estão no Exemplo 4.6, pag. 98 do livro Demétrio, C. G. B. (2002), *Modelos Lineares Generalizados em Experimentação Agronômica*, ESALQ (https://docs.ufpr.br/~niveam/micro%20da%20sala/bom/Apostila%20de%20MLG.pdf).

Adotando a função de ligação logarítmica (*default*), o preditor linear η é dado por

$$\eta = \log(\mu) = \log(\lambda) + \log(x) = \beta_1 + 1 \times \log(x), \tag{2}$$

em que $\beta_1 = \log(\lambda)$. O coeficiente do termo $\log(x)$ no preditor linear é constante ($\beta_2 = 1$), de modo que $\log(x)$ representa um deslocamento (*offset*) no preditor linear.

```
# Separador decimal nos resultados: ","
options(OutDec = ",")
```

```
# Dados (Tabela 4, p. 21)
contagens <- c(13, 14, 17, 22, 9, 14, 6, 14, 4, 4, 3, 5, 3, 2, 1, 3,
               2, 1, 3, 2, 2)
dil <- rep(c(0.3162, 0.1778, 0.1, 0.0562, 0.0316), times = c(4, 4, 4, 4,  5))
dilf <- factor(rep(c("d1", "d2", "d3", "d4", "d5"), times = c(4, 4, 4, 4, 5)))
```

O vetor `dilf` corresponde aos valores de diluição em forma de fator (variável qualitativa) para possibilitar o ajuste de um modelo com diferentes interceptos.

Modelos Poisson com diferentes interceptos e com intercepto único serão ajustados com a função `gamlss` do pacote `gamlss` em linguagem R (ou seja, a função `glm` não será usada).

```
## Modelo com diferentes interceptos
m1 <- gamlss(contagens ~ dilf + offset(log(dil)), family = PO)
```

```
## GAMLSS-RS iteration 1: Global Deviance = 82,0262
## GAMLSS-RS iteration 2: Global Deviance = 82,0262
```

```
summary(m1)
```

```
## ******************************************************************
## Family:  c("PO", "Poisson")
##
```

```
## Call:  gamlss(formula = contagens ~ dilf + offset(log(dil)),
##     family = PO)
##
## Fitting method: RS()
##
## ------------------------------------------------------------------
## Mu link function:  log
## Mu Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3,9547     0,1231  32,128 5,83e-16 ***
## dilfd2        0,1473     0,1960   0,751    0,463
## dilfd3       -0,2659     0,2787  -0,954    0,354
## dilfd4       -0,2650     0,3553  -0,746    0,467
## dilfd5        0,1930     0,3393   0,569    0,577
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## ------------------------------------------------------------------
## No. of observations in the fit:  21
## Degrees of Freedom for the fit:  5
##       Residual Deg. of Freedom:  16
##                       at cycle:  2
##
## Global Deviance:     82,0262
##             AIC:     92,0262
##             SBC:     97,24881
## ******************************************************************
```

Nota 1. Diferentes interceptos são necessários?

Abaixo apresentamos a matriz do modelo $\boldsymbol{X}$ correspondente ao modelo com diferentes interceptos.

```
model.matrix(m1)
```

```
##    (Intercept) dilfd2 dilfd3 dilfd4 dilfd5
## 1            1      0      0      0      0
## 2            1      0      0      0      0
## 3            1      0      0      0      0
## 4            1      0      0      0      0
## 5            1      1      0      0      0
## 6            1      1      0      0      0
## 7            1      1      0      0      0
## 8            1      1      0      0      0
## 9            1      0      1      0      0
## 10           1      0      1      0      0
## 11           1      0      1      0      0
## 12           1      0      1      0      0
## 13           1      0      0      1      0
## 14           1      0      0      1      0
## 15           1      0      0      1      0
## 16           1      0      0      1      0
## 17           1      0      0      0      1
## 18           1      0      0      0      1
## 19           1      0      0      0      1
## 20           1      0      0      0      1
## 21           1      0      0      0      1
```

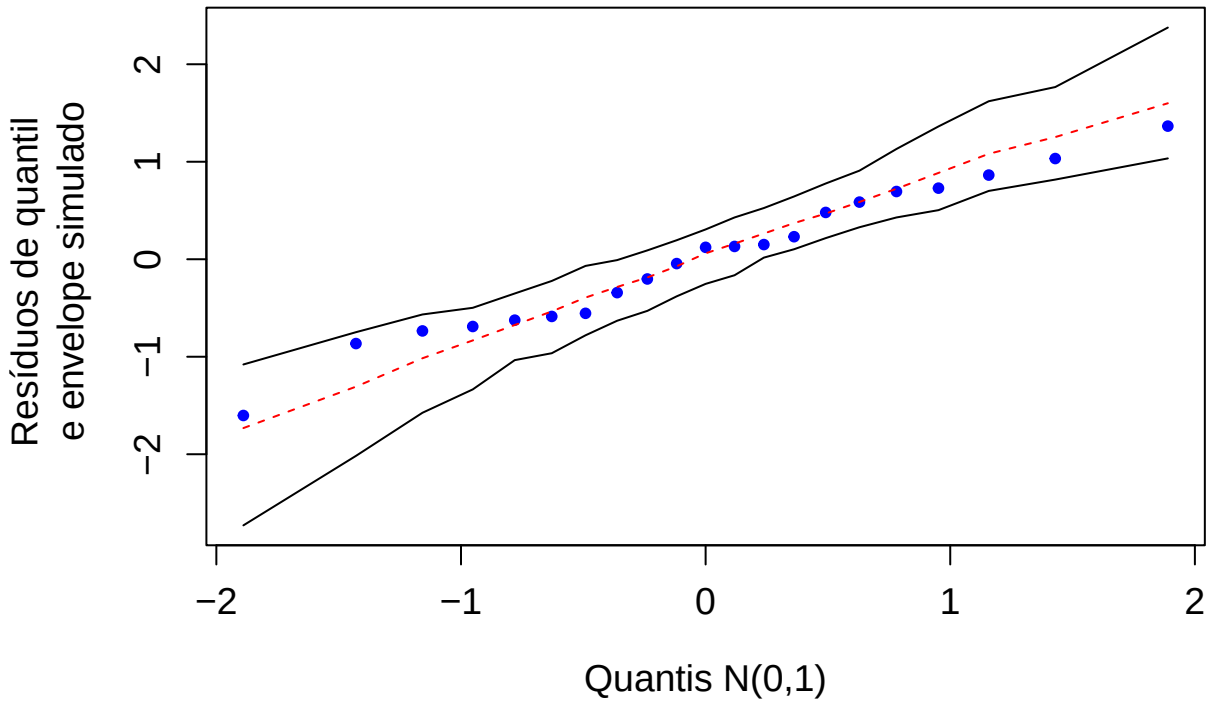
```
## attr(,"assign")
## [1] 0 1 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$dilf
## [1] "contr.treatment"
```

Em seguida apresentamos o gráfico de resíduos de quantil com envelope para o modelo com diferentes interceptos.

```
## Número de simulações
B <- 100

rq <- resid(m1)
rqo <- sort(rq)

# Simulações
set.seed(1890)
n <- length(contagens)

mrq <- matrix(0, B, n)
for (b in 1:B) {
   ysim <- rPO(n, mu = m1$mu.fv)
   msim <- gamlss(ysim ~ dilf + offset(log(dil)), family = PO)
   rqs <- resid(msim)
   mrq[b,] <- rqs
}
```

```
mrq <- t(apply(mrq, 1, sort))
Z <- qnorm((1:n - 3/8) / (n + 1/4))
rqm <- apply(mrq, 2, mean)
rq25 <- apply(mrq, 2, function(x) quantile(x, 0.025))
rq975 <- apply(mrq, 2, function(x) quantile(x, 0.975))
mrq <- cbind(Z, rqo, rq25, rqm, rq975)
```

```
# Envelope
par(mai = c(1.2, 1.2, 0.5, 0.1))
plot(mrq[, 1], mrq[, 2], pch = 20, ylim = range(mrq[, -1]),
     cex.axis = 1.2, cex.lab = 1.2, xlab = "Quantis N(0,1)",
     ylab = "Resíduos de quantil \n e envelope simulado", col = "blue")
lines(mrq[, 1], mrq[, 3])
lines(mrq[, 1], mrq[, 4], lty = 2, col = "red")
lines(mrq[, 1], mrq[, 5])
```




```
## Modelo com intercepto único
m2 <- gamlss(contagens ~ offset(log(dil)), family = PO)
```

```
## GAMLSS-RS iteration 1: Global Deviance = 85,194
## GAMLSS-RS iteration 2: Global Deviance = 85,194
```

```
summary(m2)
```

```
## ******************************************************************
## Family:  c("PO", "Poisson")
##
## Call:  gamlss(formula = contagens ~ offset(log(dil)), family = PO)
##
## Fitting method: RS()
##
## ------------------------------------------------------------------
## Mu link function:  log
## Mu Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3,95502    0,08333   47,46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
##
## ------------------------------------------------------------------
## No. of observations in the fit:  21
## Degrees of Freedom for the fit:  1
##       Residual Deg. of Freedom:  20
##                       at cycle:  2
##
## Global Deviance:     85,19397
##             AIC:     87,19397
##             SBC:     88,2385
```

```
## ********************************************************************
```

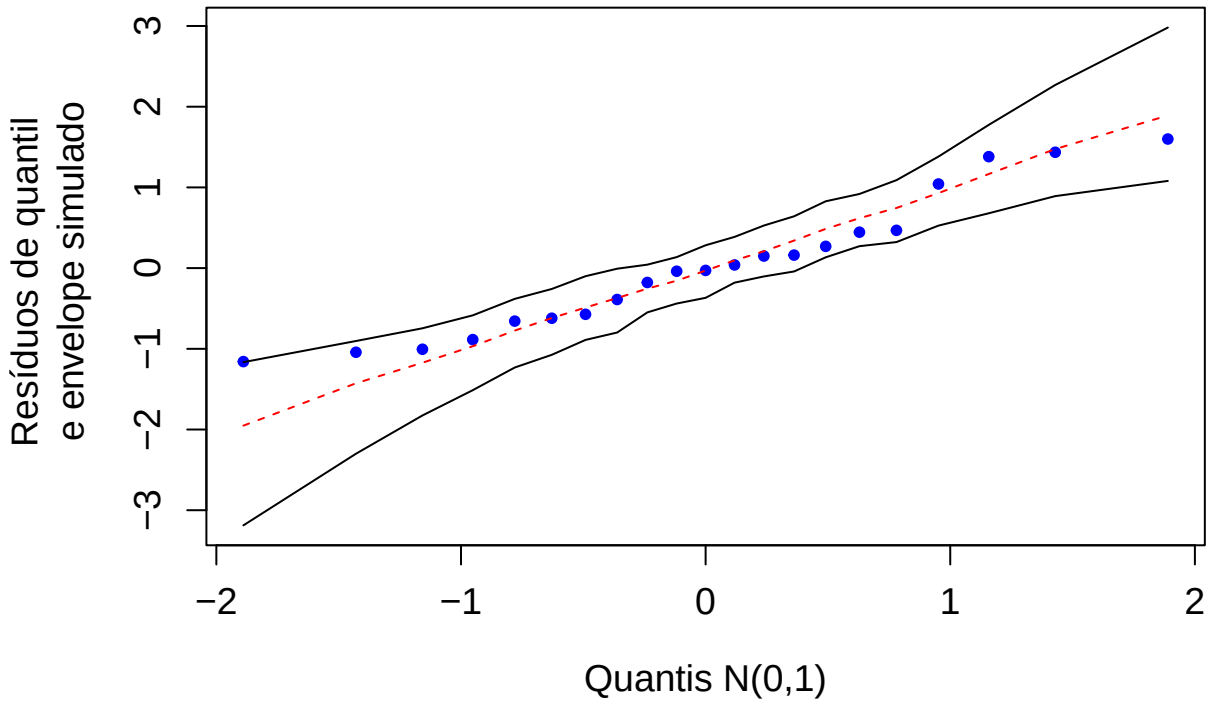
Nota 2. Procure interpretar as estimativas dos coeficientes da regressão ($\boldsymbol{\beta}$).

Em seguida apresentamos o gráfico de resíduos de quantil com envelope para o modelo com intercepto único.

```
rq <- resid(m2)
rqo <- sort(rq)

# Simulações
n <- length(contagens)

mrq <- matrix(0, B, n)
for (b in 1:B) {
   ysim <- rPO(n, mu = m2$mu.fv)
   msim <- gamlss(ysim ~ offset(log(dil)), family = PO)
   rqs <- resid(msim)
   mrq[b,] <- rqs
}
```

```
mrq <- t(apply(mrq, 1, sort))
Z <- qnorm((1:n - 3/8) / (n + 1/4))
rqm <- apply(mrq, 2, mean)
rq25 <- apply(mrq, 2, function(x) quantile(x, 0.025))
rq975 <- apply(mrq, 2, function(x) quantile(x, 0.975))
mrq <- cbind(Z, rqo, rq25, rqm, rq975)
```

```
# Envelope
par(mai = c(1.2, 1.2, 0.5, 0.1))
plot(mrq[, 1], mrq[, 2], pch = 20, ylim = range(mrq[, -1]),
     cex.axis = 1.2, cex.lab = 1.2, xlab = "Quantis N(0,1)",
     ylab = "Resíduos de quantil \n e envelope simulado", col = "blue")
lines(mrq[, 1], mrq[, 3])
lines(mrq[, 1], mrq[, 4], lty = 2, col = "red")
lines(mrq[, 1], mrq[, 5])
```



Os gráficos de resíduos com envelopes não indicam mal ajuste dos modelos. Em seguida testamos os dois modelos encaixados utilizando a estatística de teste da razão de verossimilhanças, lembrando que `G.deviance` significa $-2\ell(\widehat{\boldsymbol{\beta}}; \boldsymbol{y})$.

```
trv <- m2$G.deviance - m1$G.deviance
glt <- m2$df.residual - m1$df.residual
cat("\n Estatística de teste:", trv, ", gl:", glt,
    "(p =", pchisq(trv, glt, lower.tail = FALSE),")")
```

```
##
##  Estatística de teste: 3,167772 , gl: 4 (p = 0,5301521 )
```

Com base no resultado acima, selecionamos o modelo com intercepto único. Abaixo apresentamos uma estimativa pontual do número médio de partículas de vírus por unidade de volume.

```
cat("Número médio de partículas =", exp(coef(m2)), "\n")
```

```
## Número médio de partículas = 52,19661
```

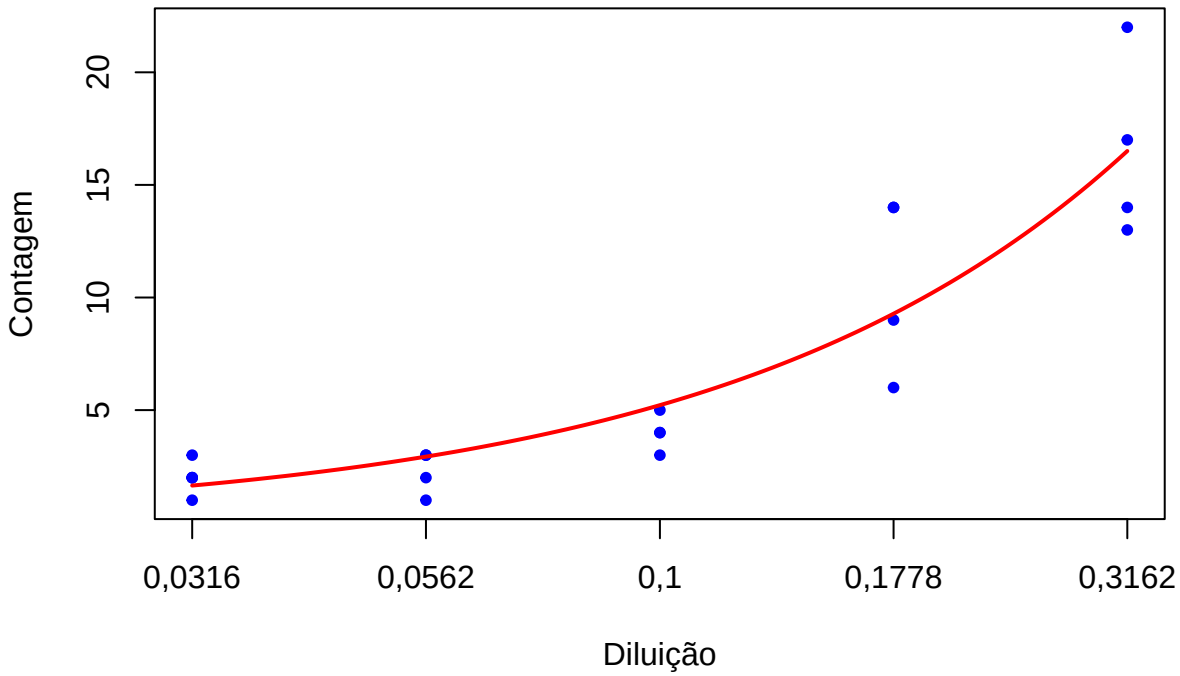
Nota 3. Apresente um intervalo de confiança para o número médio de partículas de vírus por unidade de volume.

Os dados e o modelo ajustado para μ são representados na figura a seguir, notando que no eixo horizontal a escala é logarítmica. A função `m2func` representa

$$\widehat{\mu} = \exp\left(\widehat{\beta}_1 + \log(x)\right) = \exp(\widehat{\beta}_1)x. \tag{3}$$

```
m2func <- function(x) {
   return(exp(coef(m2)) * x)
}

plot(dil, contagens, pch = 20, col = "blue", xlab = "Diluição",
     ylab = "Contagem", axes = FALSE, log = "x")
axis(1, unique(dil), unique(dil))
```

```r
axis(2)
curve(m2func, add = TRUE, col = "red", lwd = 2)
box()
```



Nota 4. Refaça o exemplo com a função de ligação identidade.

Nota 5. Refaça o exemplo em linguagem Python.