

Lista 4

Modelos Lineares Generalizados - 2/2023

César Augusto Galvão - 19/0011572

Micael Egídio Papa - 21/1029236

Questão 1

Suponha $Y_i \sim \exp(\phi)$, para $i = 1, \dots, 5$. Isto é

$$f(y_i|\phi) = \frac{1}{\phi} \exp\left(-\frac{y_i}{\phi}\right), \quad \text{com } y_i > 0 \text{ e } \phi > 0. \quad (1)$$

Suponha que o vetor observado foi $\mathbf{y} = (2.8, 3.5, 2.4, 1.9, 3.0)^\top$ com média $\bar{y} = 2.72$

Item a)

Implementa-se manualmente o algoritmo Newton-Raphson da forma

$$\beta^{(r+1)} = \beta^{(r)} + \mathcal{H}^{-1}(\beta^{(r)}) U(\beta^{(r)}) \quad (2)$$

em que $U(\beta^{(r)})$ denota o vetor escore avaliado no ponto $\beta^{(r)}$ e $\mathcal{H}(\beta^{(r)}) = -\frac{\partial^2 \ell}{\partial \beta^\top \partial \beta} \Big|_{\beta=\beta^{(r)}}$ é a matriz Hessiana.

Como se trata de uma função da família exponencial, temos que a matriz de informação esperada coincide com a matriz Hessiana

$$\mathbf{I}(\beta) \equiv \mathcal{H}(\beta)$$

,

De modo que se pode utilizar o algoritmo FS com igual resultado.

```
lik.funct<- function(y,p){n<-length(y); return(-n*log(p) - sum(y/p))} #ok
score.vec<- function(y,p){n<-length(y); return(-n/p + sum(y)/p^2)}
fisher.info<- function(y,p){n<-length(y); return(n/p^2)}

#p é o phi
fs.expo<-function(y,p,max.iter=10,eps=1e-05){

  logL.vec<-double()
  p.vec<-double()
  tol.vec<-double()
  n<-length(y)
```

```

##
p.vec[1]<-p.anter<-p
logL.vec[1]<-lik.funct(y,p.anter)
contador<-1
tol.vec[1]<-tol<-1
##
while(TRUE){

  if((p.anter>0)){
    v<-solve(fisher.info(y,p.anter),score.vec(y,p.anter))

    ## Etapa iterativa
    p.atual<- p.anter+v
  }
  else
    return(NA)

  ## Calculando a tolerância
  ## tol<- abs(p.atual-p.anter)/p.anter
  tol<- abs(lik.funct(y,p.atual)-lik.funct(y,p.anter))
  ##
  contador<-contador+1

  ## Atualizando as quandidades
  p.anter<- p.atual
  p.vec[contador]<-p.anter
  tol.vec[contador]<-tol
  logL.vec[contador]<-lik.funct(y,p.atual)
  cat("Wait: the FS algorithm is running", "\r")

  if((eps>tol)|(contador>max.iter))break

}
res<-matrix(NA,nrow = contador,ncol = 3)
colnames(res)<-c("phi.hat", "LogLik", "Tolerance")
res[,1]<-p.vec
res[,2]<-logL.vec
res[,3]<-tol.vec
##
var.phat<-solve(fisher.info(y,p.atual))

return(list(res=res, var.phat=var.phat, contador = contador))
}

resultados <- fs.expo(c(2.8, 3.5, 2.4, 1.9, 3.0), 2)

```

```
## Wait: the FS algorithm is running Wait: the FS algorithm is running
```

Utilizando um chute inicial $\lambda_0 = 2$, obtemos as seguintes estimações e a tolerância em 3 iterações

```

resultados$res %>%
  knitr::kable(
    format = "latex",

```

```

align = c("c"),
booktabs = TRUE,
longtable = TRUE,
linesep = "",
escape = FALSE
) %>%
kableExtra::kable_styling(
  position = "center",
  latex_options = c("striped", "repeat_header"),
  stripe_color = "gray!15")

```

phi.hat	LogLik	Tolerance
2.00	-10.26574	1.0000000
2.72	-10.00316	0.2625765
2.72	-10.00316	0.0000000

Item b)

Quando se compara $\hat{\phi}$ com \bar{y} , nota-se que são realmente muito próximos, com sua diferença tendendo a zero.

Questão 4

```

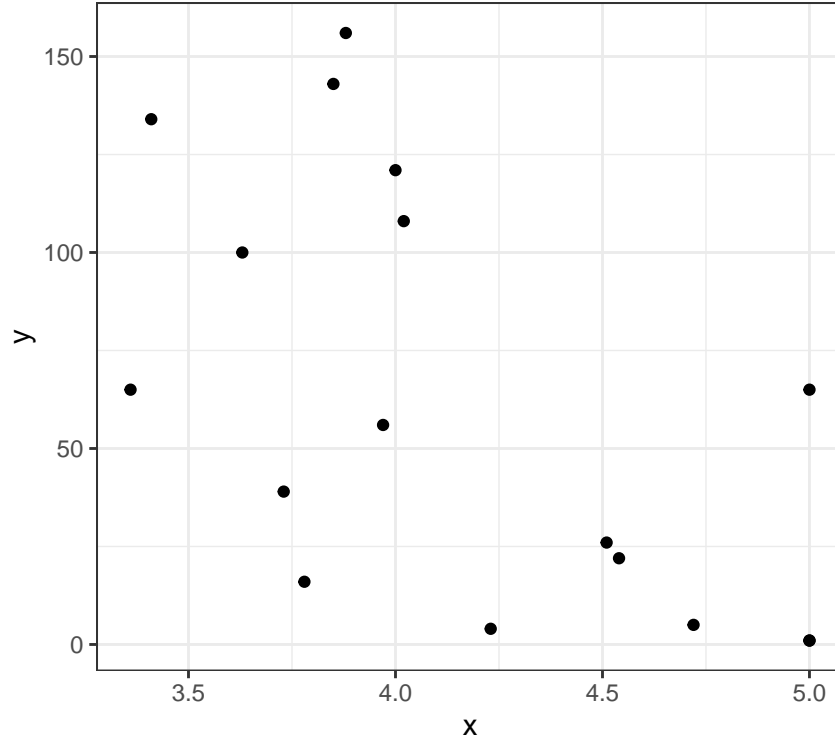
y <- c(65,156,100,134,16,108,121,04,39,143,56,26,22,1,1,5,65)

x <- c(3.36, 3.88, 3.63, 3.41, 3.78, 4.02, 4, 4.23, 3.73, 3.85, 3.97, 4.51, 4.54, 5, 5, 4.72, 5)

data <- tibble(y, x)

ggplot(data, aes(x, y))+
  geom_point()+
  theme_bw()

```



Item a)

Parece haver alguma tendência, mesmo com um pequeno tamanho de amostra, mas essa não parece ser linear. Parece haver um decaimento amortecido de y para maiores valores de x .

Item b)

Uma função de ligação apropriada parece ser o logaritmo, pois

$$\mu_i = \mathbb{E}(Y_i) = \exp \beta_0 + \beta_1 x_i, \quad \mathbf{x}_i^\top \beta = \beta_0 + \beta_1 x_i,$$

de modo que se obtém o preditor linear:

$$\log(\mu_i) = \beta_0 + \beta_1 x_i = \mathbf{x}_i^\top \beta.$$

Item c)

Reescrevemos $f(y_i; \mu) = \exp \left\{ \log(1/\mu) - y_i \frac{1}{\mu} \right\}$, de modo que a função de cumulantes fica evidentemente dada por $\log(1/\mu)$. Logo, temos

$$\mathbb{E}(Y_i) = b'(\theta) = \frac{d}{d\theta} \log \theta = \frac{1}{\theta} = \mu; \quad \theta = \frac{1}{\mu} \quad (3)$$

$$V(\mu) = b''(\theta) = -\frac{1}{\theta^2} = -\mu^2 \quad (4)$$

como $a(\phi) = -1$,

$$\text{Var}(Y_i) = a(\phi)V(\mu) = \mu^2. \quad (5)$$

Item d)

Ajusta-se um modelo de regressão exponencial, considerando que a distribuição exponencial é um caso específico da distribuição Gama.

```
fit <- glm(y ~ x, family = Gamma(link = "log"), data = data)

summary(fit)

##
## Call:
## glm(formula = y ~ x, family = Gamma(link = "log"), data = data)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.1188      1.8938   4.815 0.000227 ***
## x            -1.2424      0.4522  -2.748 0.014953 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.9752649)
##
##      Null deviance: 26.282  on 16  degrees of freedom
## Residual deviance: 20.235  on 15  degrees of freedom
## AIC: 174.75
##
## Number of Fisher Scoring iterations: 8
```

Item e)

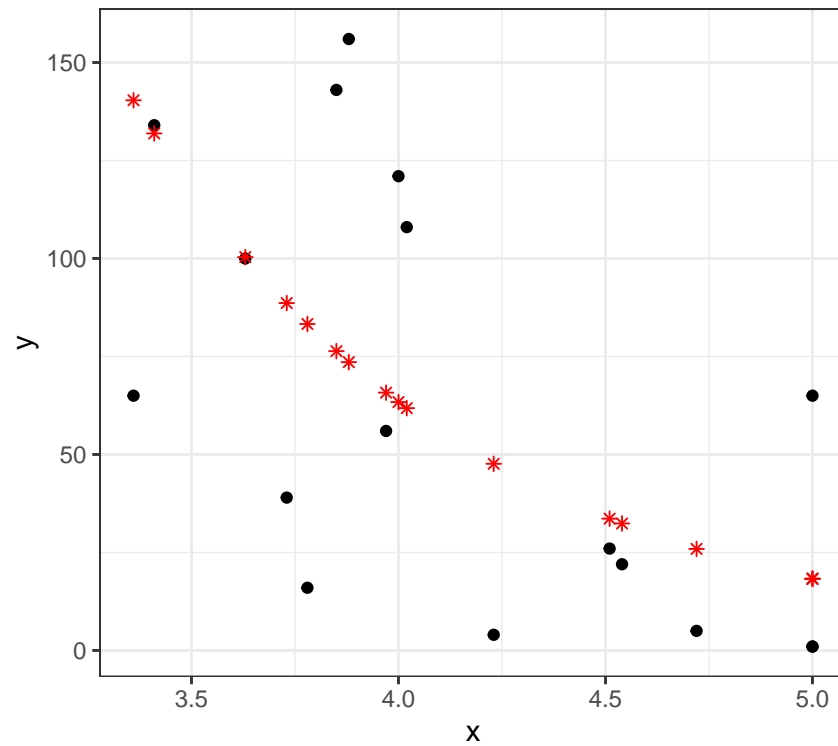
i)

A seguir são comparados graficamente os pontos observados com os estimados, os quais têm forma de asterisco. De fato, expressam a tendência de decaimento suposta no item a). No entanto, y aparenta ter uma variância ampla, se distanciando muito da média em boa parte das observações.

```
yhat <- fit$fitted.values

compara <- tibble(
  x, y, yhat, y-yhat
)

ggplot(data, aes(x, y))+
  geom_point()+
  geom_point(aes(x, yhat), data = compara, color = "red", shape = 8)+
  theme_bw()
```

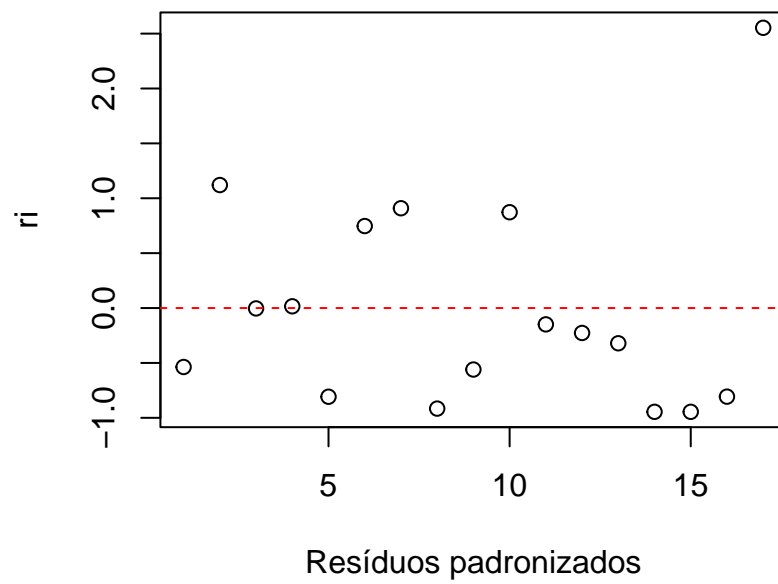


ii)

Construimos os resíduos padronizados e expomos seu gráfico de dispersão a seguir:

```
ri <- (y-yhat)/yhat

plot(ri, xlab = 'Resíduos padronizados')
abline(h = 0, lty = 2, col = 'red')
```



Os resíduos padronizados parecem bem comportados em torno de zero, com valor absoluto inferior a 2, exceto uma das observações.