Estudo sobre modelos log-lineares Poisson: Verificação de estimativa pontual e intervalar (tipo link) para as médias

André Savassi
Kenzo Bontempo
Márcio Antônio

UFMG

01/10/2025

Sumário

Introdução

Aqui vai o conteúdo do seu slide.

- Este é um item de lista.
- Outro item importante.

Título de um Bloco

Texto dentro de um bloco destacado.

Introdução

Aqui vai o conteúdo do seu slide.

- Este é um item de lista.
- Este item aparece no segundo "clique" (sobreposição).
- Outro item importante.

Título de um Bloco

Texto dentro de um bloco destacado.

Gerando amostras de distribuições - Poisson

Primeiramente, iremos amostrar da Distribuição Poisson, que possui a função densidade descrita por:

$$f(y;\lambda) = \frac{e^{-\lambda}\lambda^{y}}{y!},\ y = 0, 1, \dots, n$$

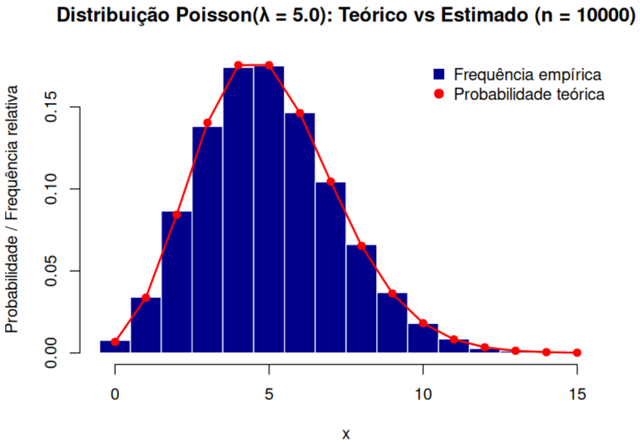
Referencia-se que: $\mathbb{E}(Y) = Var(Y) = \lambda$

Amostrador Poisson via Aceitação e Rejeição

```
poisson_ar_1 <- function(lambda) {
  f_pois <- function(k, lambda) {
    return(exp(k * log(lambda) - lambda - lgamma(k +
        1)))
  }
  fmax <- f_pois(floor(lambda), lambda)
  kmax <- qpois(0.9999, lambda = 5)
  repeat {
    y <- sample(0:kmax, 1)
    fy <- f_pois(y, lambda)
    gy <- 1 / (kmax + 1)
    u <- runif(1)
    if (u < fy / (fmax)) {
      return(y)
    }}}
poisson_ar <- function(lambda, n = 1) {
  replicate(n, poisson_ar_1(lambda))
}
```

Código.R

Gráficos - Poisson AR



Gerando amostras de distribuições - Binomial Negativa

Segue que a densidade da distribuição Binomial Negativa é descrita por:

$$f(y; r, p) = \binom{y-1}{r-1} p^r (1-p)^{y-r}$$

onde r é o total de sucessos e p é a probabilidade desses sucessos. Note que $Y \sim BN(r, p)$ possui esperança e variância a seguir:

$$\mathbb{E}(Y) = \frac{r}{p} \;;\; Var(Y) = \frac{r(1-p)}{p^2}$$

Amostrador BN via Amostragem por Importância

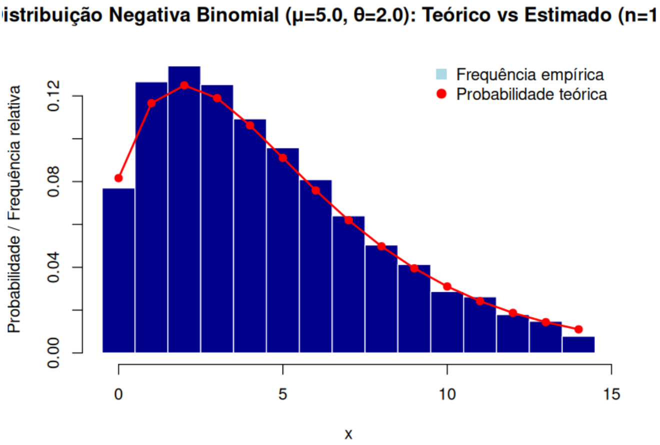
```
rnegbinom <- function(m, n, mu, theta) {
  f_negbinom <- function(x, mu, theta) {

    log_f <- lgamma(x + theta) -
    lgamma(theta) -
    lgamma(x + 1) +
    theta * log(theta / (mu + theta)) +
    x * log(mu / (mu + theta))
    return(exp(log_f))
  }

  g_pois <- function(k, lambda) {
    return(exp(k * log(lambda) - lambda - lgamma(k +
          1)))
  }
```

Código.R

Amostrador BN via Amostragem por Importância

```
source("rpoisson.R")

y <- poisson_ar(n = m, lambda = 2 * mu)

pesos <- f_negbinom(y, mu = mu, theta = theta) / g_pois(y,
    lambda = 2 * mu)
pesos <- pesos / sum(pesos)

amostra <- sample(y, n, F, pesos)

return(amostra)
}
```

Gráficos - BN Amostragem por Importância



Gerando amostras de distribuições - Bell

Segue que a densidade da distribuição Bell é descrita por:

$$f(y;\theta) = \frac{\theta e^{e^{\theta}+1}}{y!} B_Y, \; y = 0, 1, \ldots, n; \; \theta > 0$$

Onde o termo B_Y corresponde ao número de Bell, dados por:

$$B_n = \frac{1}{e} \sum_{k=0}^{\infty} \frac{k^n}{k!}, \text{iniciando com } B_0 = B_1 = 1$$

A média e variância de $Y \sim Bell(\theta)$ é:

$$\mathbb{E}(Y) = \theta e^{\theta} \; ; \; Var(Y) = \theta(1+\theta)e^{\theta}$$

Amostrador Bell via Transformação Inversa

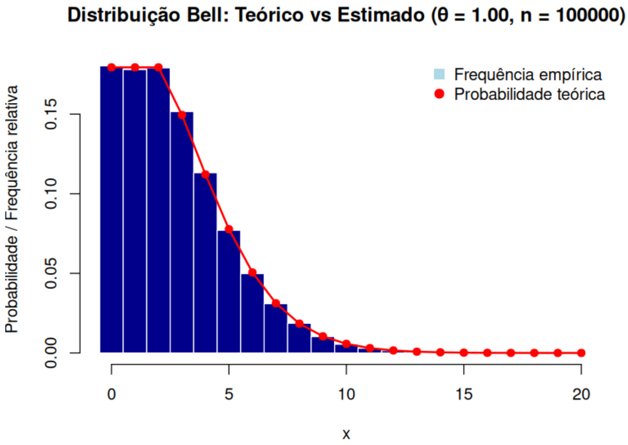
```r
# Importado de bellreg
dbell <- function(x, theta, log = FALSE) {
  Bx <- c()
  for (i in 1:length(x)) {
    Bx[i] <- numbers::bell(x[i])
  }
  lf <- x * log(theta) - exp(theta) + 1 + log(Bx) -
       lgamma(x + 1)
  if (log == TRUE) {
    return(lf)
  } else {
    return(exp(lf))
  }
}
```

▸ Código.R

Amostrador Bell via Transformação Inversa

```
rbell_aux <- function(theta) {
  sapply(theta, function(t) {
    u <- runif(1, 0, 1)
    i <- 0
    pr <- dbell(0, t); Fx <- pr
    while (u >= Fx) {
      i <- i + 1
      pr <- dbell(i, t)
      Fx <- Fx + pr}
    return(i)
  })}
rbell <- function(n, theta) {
  if (length(theta) == 1) {
    return(replicate(n, expr = rbell_aux(theta), simplify =
        TRUE))
  } else {
    return(rbell_aux(theta))
  }}
```

Gráficos - Bell Inversa



Monte Carlo - Poisson

```
MC <- 1000
nsizes <- c(50, 100, 500, 1000)
source("rpoisson.R")
set.seed(9999)
df_teste <- data.frame(
x1 = sample(c(0, 1), 10, replace = T, prob = c(0.7,
     0.3)),
x2 = rnorm(10),
x3 = rnorm(10),
x4 = sample(c(0, 1), 10, replace = T, prob = c(0.2,
     0.8)))
run_MC <- function(r, teste, nsize) {
  set.seed(r)
  n <- nsize
  betas <- c(1.2, 0.25, -0.08, 0.15, -0.12)
  X_teste <- model.matrix(~., teste)
  eta_teste_real <- X_teste %*% betas
  fitted_teste_real <- exp(eta_teste_real)
```

Código.R

Monte Carlo - Poisson

```
MC <- 1000
nsizes <- c(50, 100, 500, 1000)
source("rpoisson.R")
set.seed(9999)
df_teste <- data.frame(
x1 = sample(c(0, 1), 10, replace = T, prob = c
    (0.7, 0.3)),
x2 = rnorm(10),
x3 = rnorm(10),
x4 = sample(c(0, 1), 10, replace = T, prob = c
    (0.2, 0.8)))
run_MC <- function(r, teste, nsize) {
  set.seed(r)
  n <- nsize
  betas <- c(1.2, 0.25, -0.08, 0.15, -0.12)
  X_teste <- model.matrix(~., teste)
  eta_teste_real <- X_teste %*% betas
  fitted_teste_real <- exp(eta_teste_real)
```

▸ Código.R

Código

```
run_MC <- function(r, nsize) {
  set.seed(r)
  n <- nsize
  betas <- c(1.2, 0.25, -0.08, 0.15, -0.12)

  df <- data.frame(
  x1 = sample(c(0, 1), n, replace = T, prob = c(0.7, 0.3)),
  x2 = rnorm(n),
  x3 = rnorm(n),
  x4 = sample(c(0, 1), n, replace = T, prob = c(0.2, 0.8))
  )
  X <- model.matrix(~., df)
  eta <- X %*% betas
  mu <- exp(eta)
```

Tamanho da Amostra	Parâmetro	Valor Real	Estimativa Média	Erro Padrão	Desvio Padrão	Viés Relativo	Limite Inferior	Limite Superior	Cobertura
100	y1	3.362	3.354	0.014	0.386	-0.225	2.673	4.211	0.950
100	y10	3.798	3.790	0.010	0.378	-0.210	3.119	4.607	0.958
100	y2	3.491	3.492	0.020	0.481	0.029	2.664	4.581	0.952
100	y3	5.447	5.497	0.025	0.914	0.934	4.036	7.494	0.937
100	y4	2.513	2.507	0.014	0.285	-0.259	1.998	3.146	0.952
100	y5	3.980	3.997	0.029	0.682	0.424	2.873	5.565	0.949
100	y6	4.220	4.217	0.011	0.440	-0.057	3.447	5.162	0.950
100	y7	3.097	3.093	0.012	0.334	-0.151	2.501	3.825	0.956
100	y8	4.512	4.514	0.021	0.658	0.039	3.401	5.998	0.959
100	y9	3.845	3.840	0.011	0.392	-0.148	3.143	4.692	0.958