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:

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

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

Amostrador Poisson via Transformação Inversa

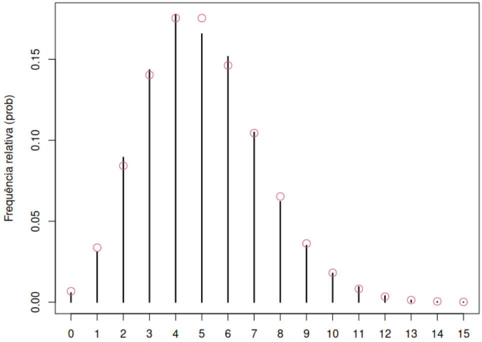
```
library(dplyr)
library(kableExtra)
rpois_aux <- function(lambda) {
  u <- runif(1, 0, 1)
  i <- 0
  pr <- exp(-lambda)
  Fx = pr
  while (u >= Fx) {
    pr <- pr * lambda / (i + 1)
    Fx <- Fx + pr
    i <- i + 1
  }
  return(i)
}
rpoisson <- function(n, lambda) {
  replicate(n, expr = rpois_aux(lambda), simplify =
      TRUE)
}
```

Código.R

Resultados - Poisson Inversa

Comparação de frequências relativas com probabilidades teóricas

	0	1	2	3	4	5	6	7	8	9
freq	0.006	0.031	0.089	0.144	0.178	0.166	0.152	0.105	0.062	0.035
prob	0.007	0.034	0.084	0.140	0.175	0.175	0.146	0.104	0.065	0.036



Amostrador Poisson via Aceitação/ Rejeição

```
poisson_ar_1 <- function(lambda, kmax = 30) {
  f_pois <- function(k, lambda) {
    return(exp(-lambda) * lambda^k / factorial(k))
  }
  fmax <- f_pois(floor(lambda), lambda)
  repeat {
    y <- sample(0:kmax, 1)
    fy <- f_pois(y, lambda)
    gy <- 1 / (kmax + 1)
    u <- runif(1)
    if (u < fy / (fmax * gy)) {
      return(y)
    }
  }
}
poisson_ar <- function(lambda, n = 1000, kmax = 30) {
  replicate(n, poisson_ar_1(lambda, kmax))
}
```

Código.R

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:

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

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 variâncida de $Y \sim Bell(\theta)$ é:

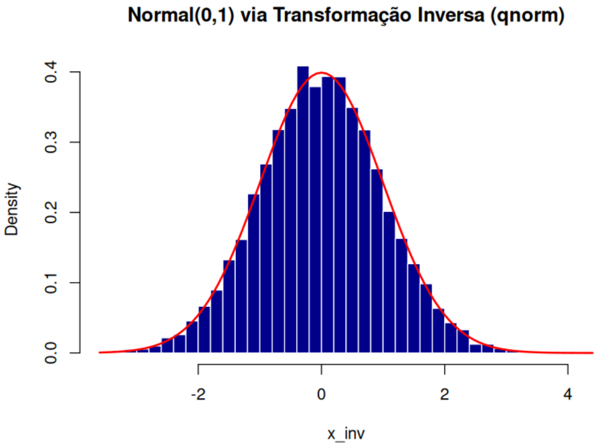
$$\mathbb{E}(Y) = \theta e^{\theta} \; ; \; Var(Y) = \theta(1+\theta)e^{\theta}$$

Amostrador Bell via Transformação Inversa

```
n <- 10000

u <- runif(n)

x_inv <- qnorm(u, mean = 0, sd = 1)
```

▸ Código.R

Gráficos - Bell Inversa

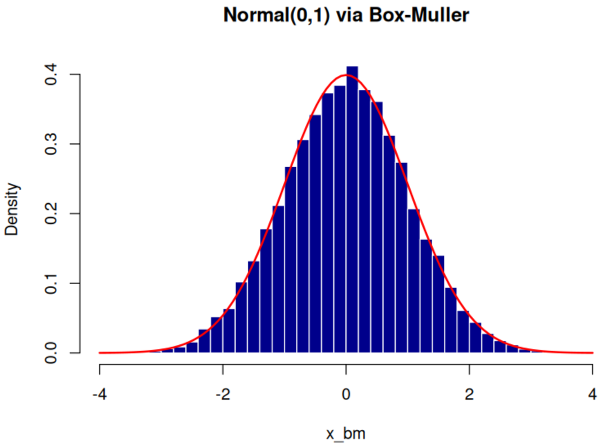


Amostrador Bell via Box Muller

```r
n <- 10000

u1 <- runif(n)
u2 <- runif(n)

z1 <- sqrt(-2 * log(u1)) * cos(2 * pi * u2)
z2 <- sqrt(-2 * log(u1)) * sin(2 * pi * u2)

x_bm <- c(z1, z2)
```

Código.R

Gráficos - Bell Box Muller



Amostrador Bell via Transformação Inversa

```r
MC <- 1000
nsize <- 100
library(foreach)
library(dplyr)
library(tibble)
library(doParallel)
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))
)
X_teste <- model.matrix(~., df_teste)
results <- data.frame()
```

Código.R

```r
run_M
  set
  n <
  bet

  df
  x1

  x2
  x3
  x4
  )
  X <
  eta
  mu
```

Gráficos - Bell Inversa