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

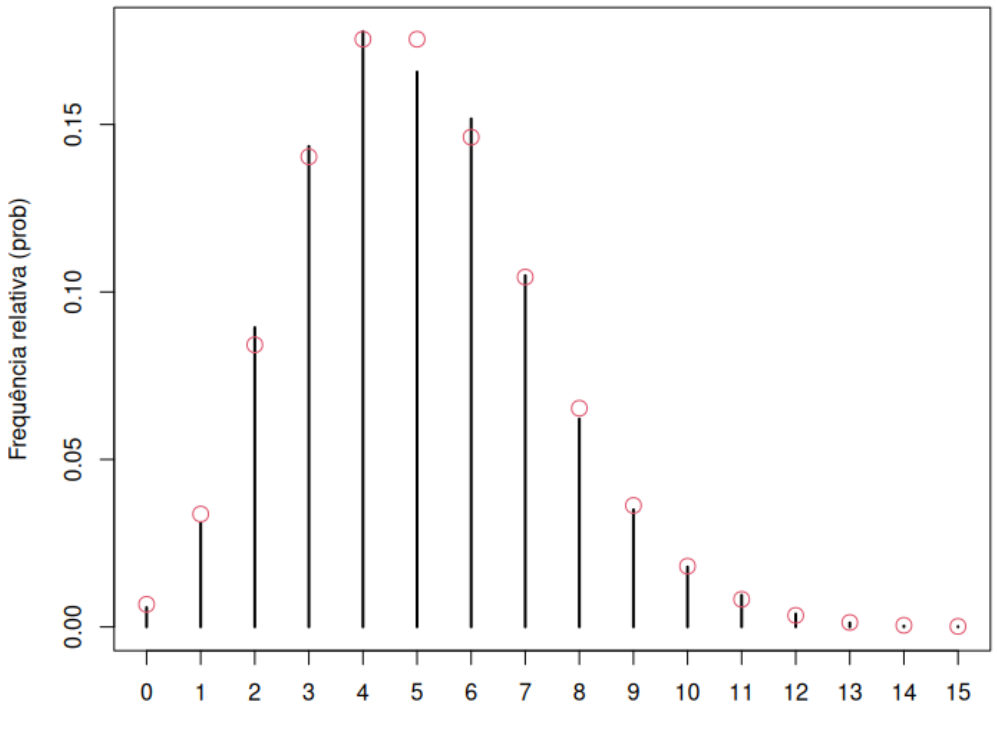
```r
library(dplyr)
library(kableExtra)
rpois_aux <- function(lambda) {
  u <- runif(1, 0, 1)
  i <- 0
  pr <- exp(-lambda)
  Fx = pr
  while (u >= Fx) {
    pr <- pr * lambda / (i + 1)
    Fx <- Fx + pr
    i <- i + 1
  }
  return(i)
}
rpoisson <- function(n, lambda) {
  replicate(n, expr = rpois_aux(lambda),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

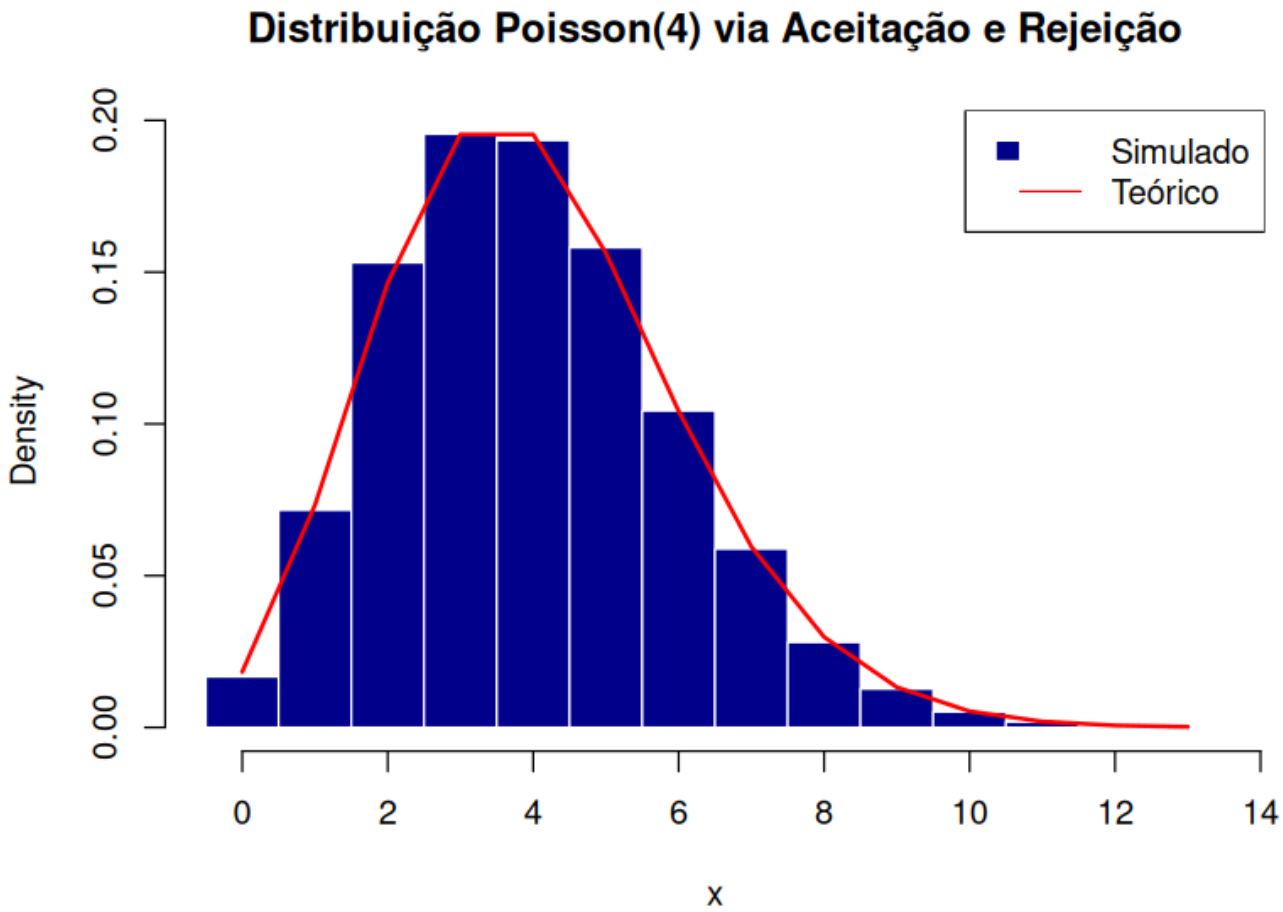


Amostrador Poisson via Aceitação/ Rejeição

```r
set.seed(20252)
rpois_ar <- function(lambda) {
  p <- 1 / (1 + lambda)
  f <- function(k) exp(-lambda) * lambda^k /
      factorial(k)
  g <- function(k) p * (1 - p)^k
  ks <- 0: (lambda*10 + 10)
  M <- max(f(ks) / g(ks))
  repeat {
    X <- rgeom(1, prob = p)
    U <- runif(1)
    if (U <= f(X) / (M * g(X))) {
      return(X)
    }
  }
}
```

▸ Código.R

Gráficos - Poisson Aceitação e Rejeição



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:

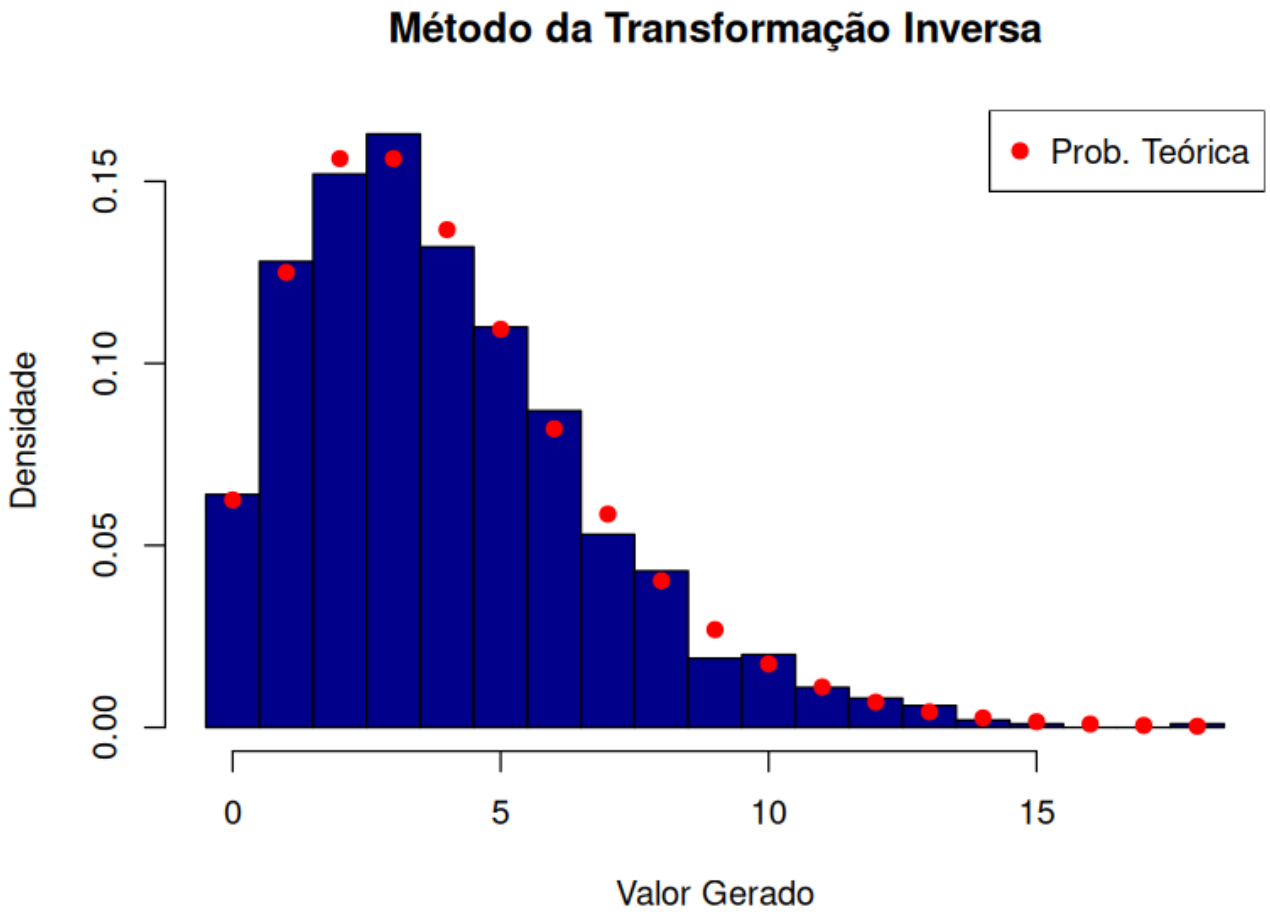
$$\mathbb{E}(Y) = \frac{r}{p} \; ; \; Var(Y) = \frac{r(1-p)}{p^2}$$

Amostrador BN via Transformação Inversa

```r
r_param <- 4
p_param <- 0.5
n_amostra <- 1000
gerar_bn_inversa_aux <- function(r, p) {
  u <- runif(1)
  i <- 0
  pr <- p^r
  Fx <- pr
  while (u >= Fx) {
    pr <- pr * (i + r) / (i + 1) * (1 - p)
    Fx <- Fx + pr
    i <- i + 1
  }
  return(i)
}
gerar_bn_inversa <- function(n, r, p) {
  return(replicate(n, gerar_bn_inversa_aux(r, p)))
}
```

▸ Código.R

Gráficos - BN Transformação Inversa

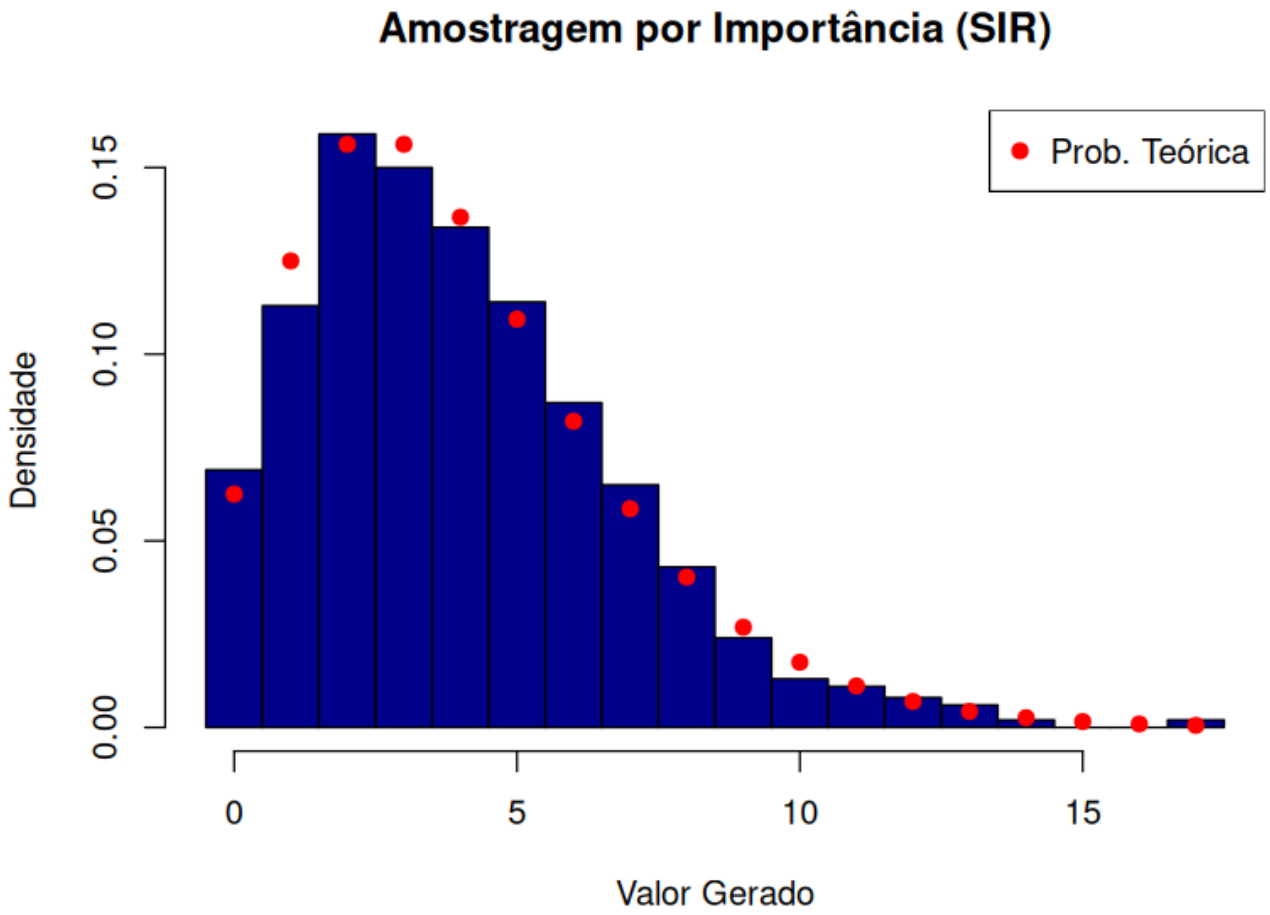


Amostrador BN via Amostragem por Importância

```r
r_param <- 4
p_param <- 0.5
n_amostra <- 1000
m_candidatos <- 20000
gerar_bn_sir <- function(n, r, p, m) {
  media_bn <- r * (1 - p) / p
  p_g <- 1 / (1 + media_bn)
  candidatos <- rgeom(m, prob = p_g)
  pesos_proposta <- dgeom(candidatos, prob = p_g)
  pesos_importancia <- pesos_alvo / pesos_proposta
  amostra_final <- sample(
  x = candidatos,
  size = n,
  replace = TRUE,
  prob = pesos_importancia
  )
  return(amostra_final)
}
```

▸ Código.R

Gráficos - BN Amostragem por Importância



Amostrador BN via Aceitação e Rejeição

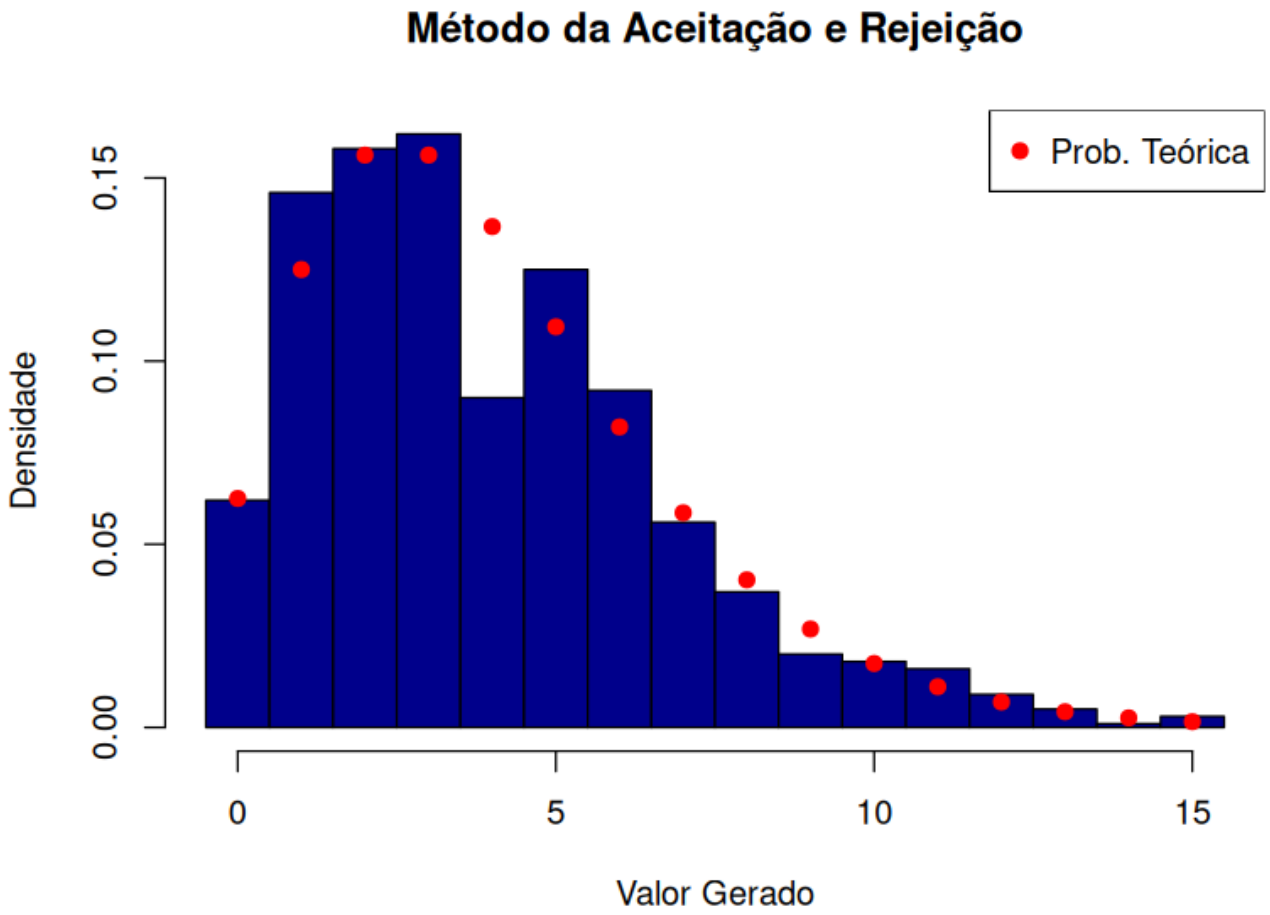
```
r_param <- 4
p_param <- 0.5
n_amostra <- 1000
gerar_bn_rejeicao <- function(n, r, p) {
  media_bn <- r * (1 - p) / p
  p_g <- 1 / (1 + media_bn)
  k_vals <- 0:100
  razao_pq <- dnbinom(k_vals, size = r, prob = p)
      / dgeom(k_vals, prob = p_g)
  c <- max(razao_pq, na.rm = TRUE) * 1.01
  amostra <- numeric(n)
  contador_aceitos <- 0
```

▸ Código.R

Amostrador BN via Aceitação e Rejeição

```
while (contador_aceitos < n) {
  w <- rgeom(1, prob = p_g)
  u <- runif(1)
  p_w <- dnbinom(w, size = r, prob = p)
  q_w <- dgeom(w, prob = p_g)
  if (u < p_w / (c * q_w)) {
    contador_aceitos <- contador_aceitos + 1
    amostra[contador_aceitos] <- w
  }
}
return(amostra)
}
```

Gráficos - BN Aceitação e Rejeição



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:

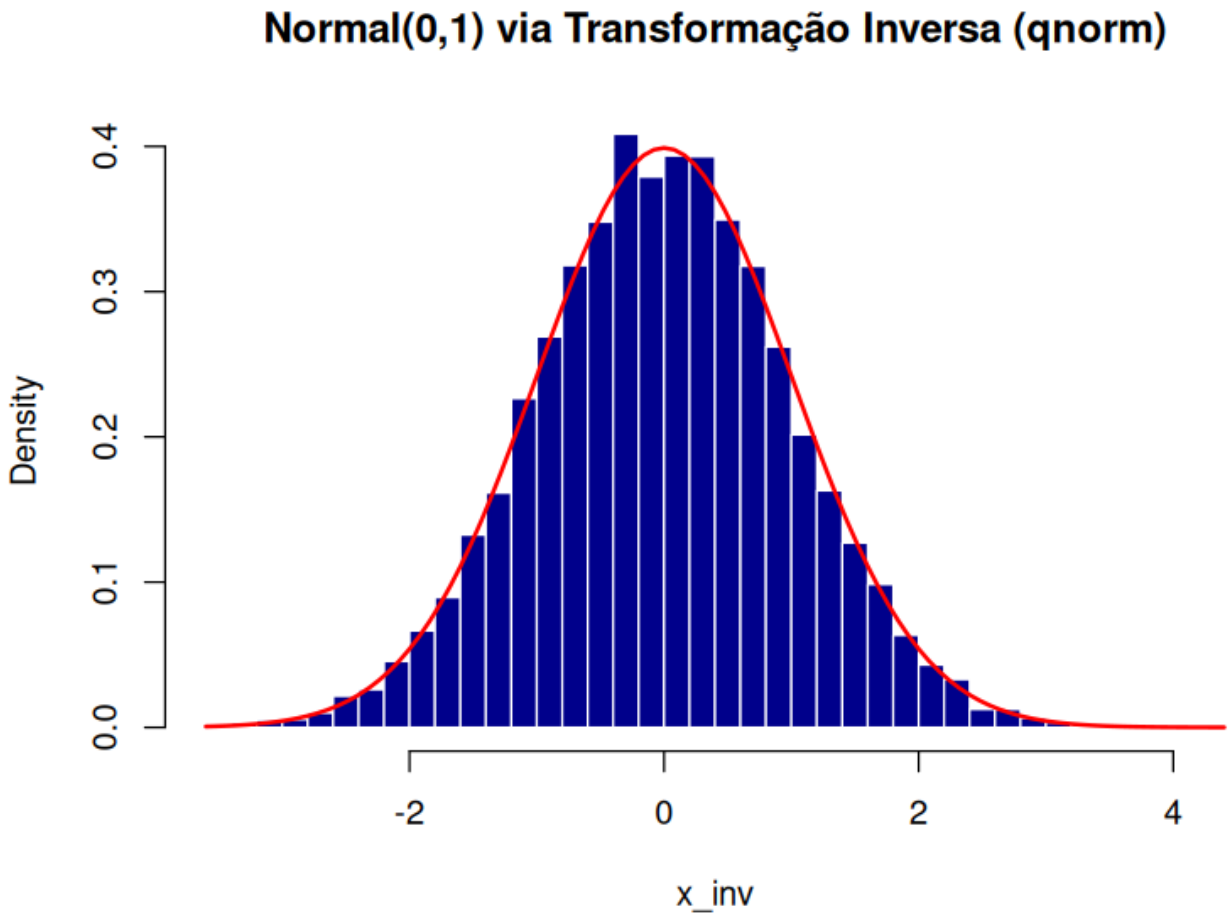
$$\mathbb{E}(Y) = \theta e^{\theta}\ ;\ Var(Y) = \theta(1+\theta)e^{\theta}$$

Amostrador Bell via Transformação Inversa

```r
n <- 10000

u <- runif(n)

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

▸ Código.R

Gráficos - Bell Inversa

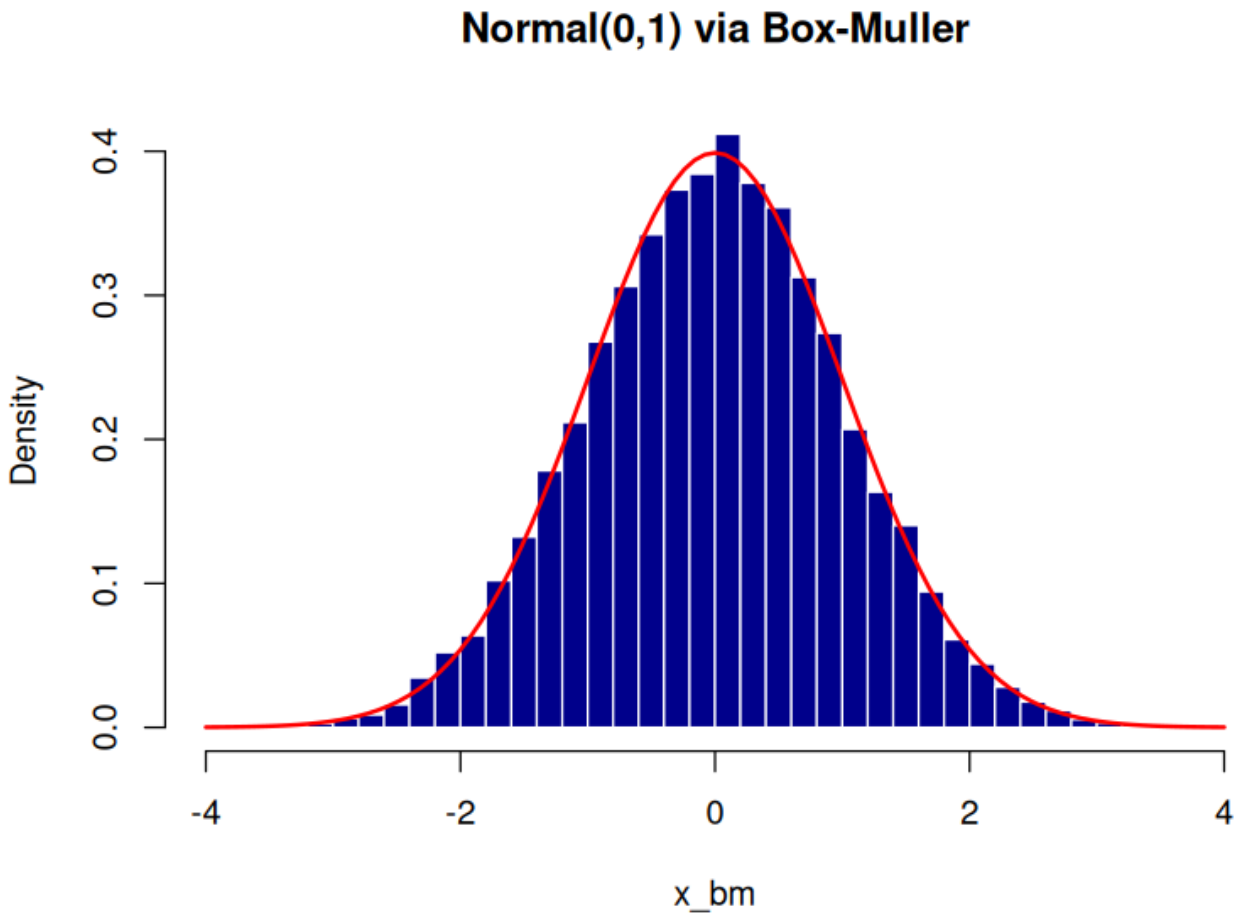


Amostrador Bell via Box Muller

```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



Código

```
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

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)
```