

Técnicas Computacionais em Estatística

Ana Paula Vanderley

2024-03-25

ALGUMAS DEMOSTRAÇÕES APLICADAS NA MATÉRIA DE TÉCNICAS COMPUTACIONAIS NO PROGRAMA DE MESTRADO EM ESTATÍSTICA

Geração de Números Aleatórios a partir da Função de Densidade dada por $f(x) = 30x^2(1-x)^2$, $0 < x < 1$

Criando uma função que calcula a densidade de probabilidade que segue a distribuição estabelecida nesse caso $60x^3(1-x)^2$, dentro do intervalo $[0,1]$.

```
Dens = function(x) {  
  dentro_intervalo = x >= 0 & x <= 1  
  densidade = numeric(length(x))  
  densidade[dentro_intervalo] = 60 * x[dentro_intervalo]^3 * (1 - x[dentro_intervalo])^2  
  return(densidade)  
}
```

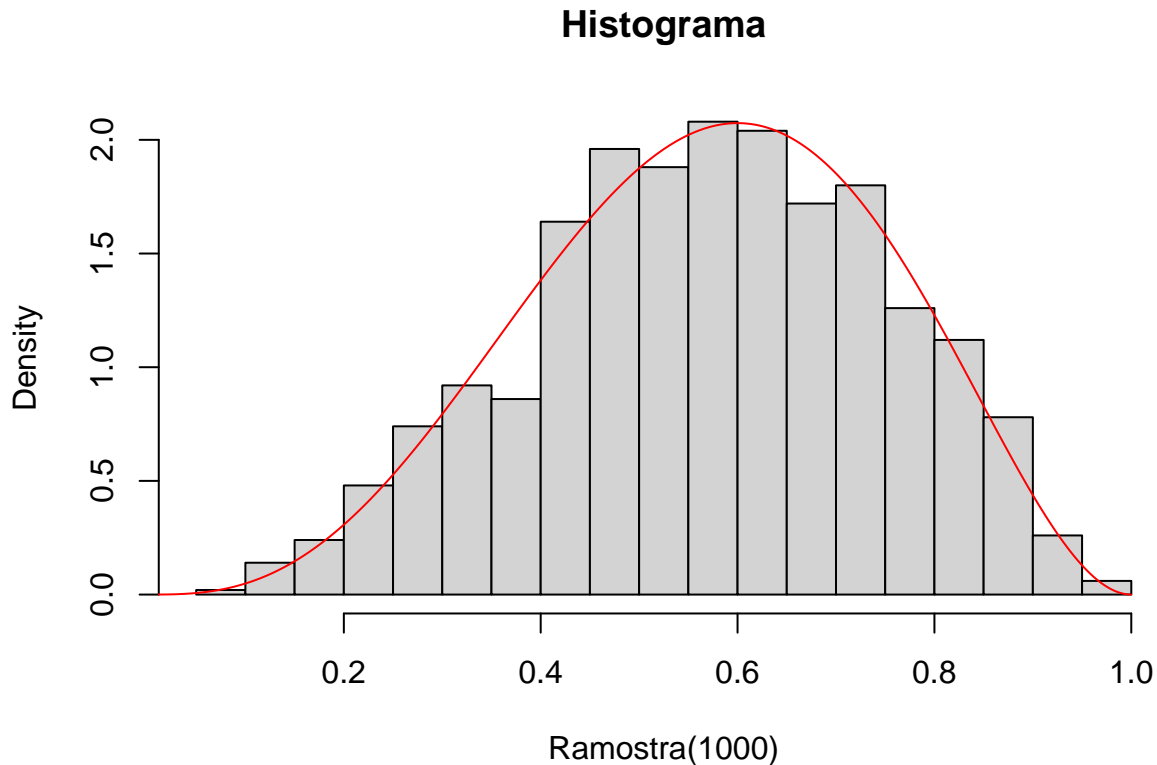
Gerando Amostras

Gerando amostras a partir do método da aceitação/rejeição o loop será executado até a amostra ser aceita, gerando valores no intervalo $[0,1]$ em x_c e verificando a densidade aplicada para o valor de x_c se a amostra gerada u_1 for menor que α a amostra é aceita.

```
Ramostra = function(n) {  
  c = 6480/3125  
  x = numeric(n)  
  
  for (i in 1:n) {  
    repeat {  
      xc = runif(1, 0, 1)  
      alpha = 1/c * 60 * (xc^3) * ((1 - xc)^2)  
      u1 = runif(1, 0, 1)  
      if (u1 < alpha) {  
        x[i] = xc  
        break  
      }  
    }  
  }  
  
  return(x)  
}
```

Plotando o Gráfico

```
hist(Ramostra(1000), breaks = 20, main = "Histograma", freq = F)
curva = seq(0, 1, length.out = 1000)
lines(curva, Dens(curva), type = "l", col="red")
```



Gerando dados aleatórios a partir da função de densidade de probabilidade $f(x) = p \lambda_1^x \exp(-\lambda_1)/x! + (1-p) \lambda_2^x \exp(-\lambda_2)/x$. Onde $\lambda > 0$.

Definindo os parametros que vou usar para gerar os amostras aleatória.

```
lambda1 <- 10
lambda2 <- 30
p <- 1/4
n <- 1000
```

Fazendo uma função que vai gerar valores binomiais aleatórios de 0 e 1 o valor 1 probabilidade p e o valor 0 probabilidade $(1-p)$ com isso determino qual das duas populações, ou seja, λ_1 e λ_2 cada observação será amostrada. E crio um vetor vazio para armazenar os valores gerados.

Fazendo um loop que irá percorrer de 1 até o número da minha amostra no caso 1000, se o valor gerado for igual a 1 a observação será gerada a partir da população com parametro λ_1 , caso contrario a observação sera gerada a partir da outra população de parametro λ_2 .

```
amostra <- function(n, lambda1, lambda2, p) {
  sample_lambda <- rbinom(n, 1, p)
  sample_x <- numeric(n)
  for (i in 1:n) {
    if (sample_lambda[i] == 1) {
      sample_x[i] <- rpois(1, lambda1)
    }
  }
}
```

```

} else {
  sample_x[i] <- rpois(1, lambda2)
}

}
return(sample_x)
}

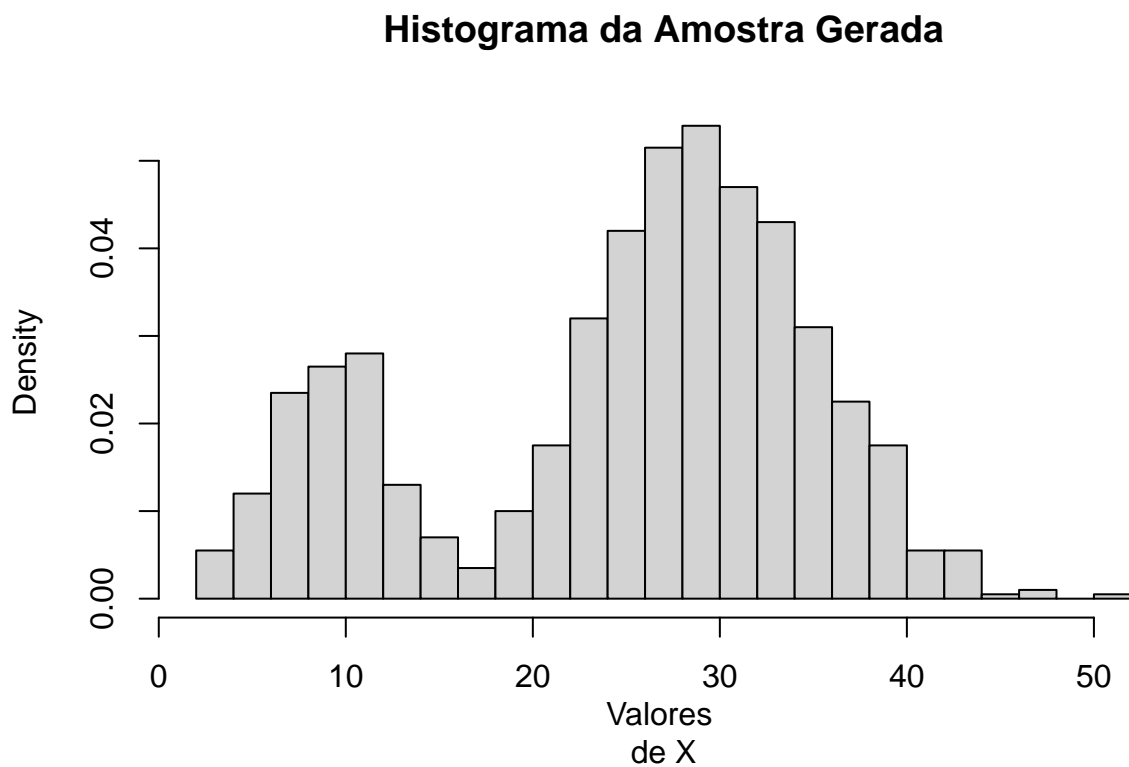
```

Gerando a amostra a partir da função criada.

```
amostra <- amostra(n, lambda1, lambda2, p)
```

Plotando o gráfico.

```
hist(amostra, breaks = 20, prob = TRUE, main = "Histograma da Amostra Gerada", xlab = "Valores de X")
```



Fazendo vetor contendo valores de 0 ao máximo da minha amostra e depois separo em 2 vetores contendo a probabilidade de cada um para plotar a linha de cada grupo.

```

x <- 0:max(amostra)
fx1 <- dpois(x, lambda1) * p
fx2 <- dpois(x, lambda2) * (1 - p)
hist(amostra, breaks = 20, prob = TRUE, main = "Histograma da Amostra Gerada", xlab = "Valores de X")
lines(x, fx1, col = "blue", lwd = 2)
lines(x, fx2, col = "red", lwd = 2)
legend("topright", legend = c(paste("Poisson(", lambda1, ")", " * p"), paste("Poisson(", lambda2, ")", " * (1 - p)")),

```

Histograma da Amostra Gerada

