

Trabalho 3

SME 0806 - Estatística Computacional

```
knitr::opts_chunk$set(message = FALSE, warning = FALSE,
                        error = FALSE, fig.dim=c(6,4), fig.pos = "H", fig.show='hold')

# Separador decimal: ","
options(OutDec = ",")
```

Introdução

Um vetor aleatório (X,Y) tem função densidade tal que

$$f(x, y) \propto x^{2,2}(1-x)^7(1-x^{0,2})^{y-1},$$

se $0 < x < 1$ e $0 < y < 1$; $f(x,y) = 0$, caso contrário.

Será apresentado um gerador de amostras do vetor (X,Y) em linguagem R.

Análise gráfica das condicionais completas

A partir da expressão de $f(x, y)$, temos que as distribuições condicionais completas têm funções densidade tais que

$$f_x(x) = f(x|y) \propto x^{2,2}(1-x)^7(1-x^{0,2})^{y-1}$$

e

$$f_y(y) = f(y|x) \propto (1-x^{0,2})^y$$

Para analisar graficamente as condicionais completas da função, adota-se $y = 0,5$ e $x = 0,5$, nos gráficos de $f_x(x)$ e $f_y(y)$ respectivamente.

```
fx = function(x,y){
  return((x^2.2)*((1-x)^7)*((1-(x^0.2))^(y-1)))
}
fy = function(y,x){
  return((1-(x^0.2))^y)
```

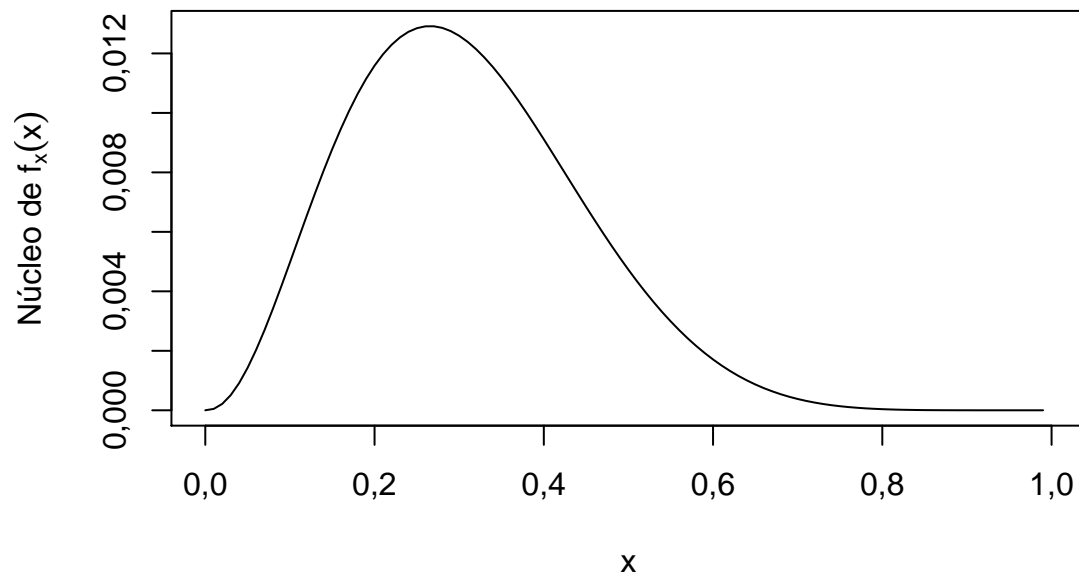
```

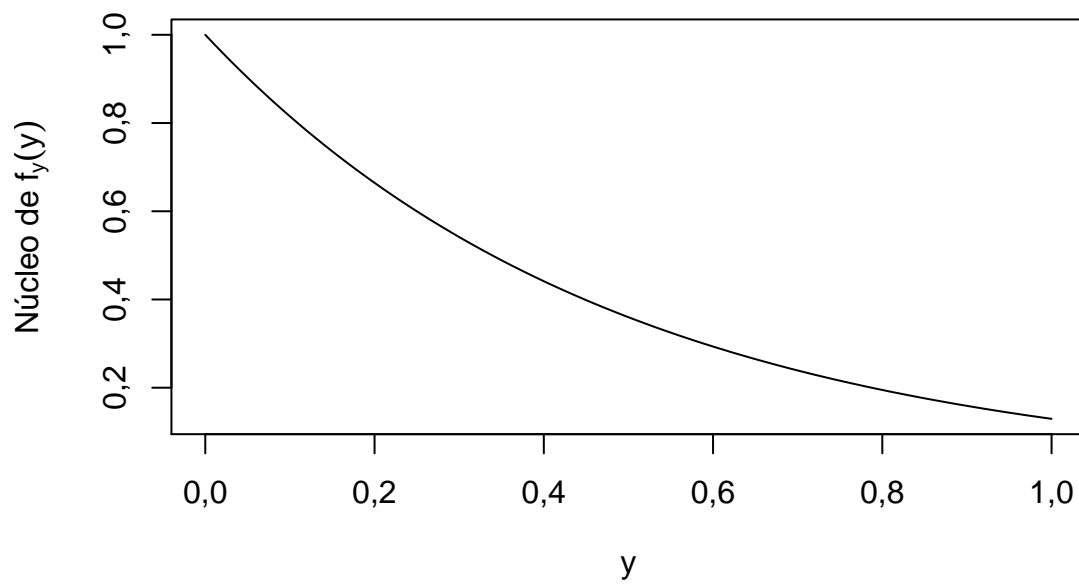
}

curve(fx(x, y = 0.5), xlab = expression(x), 0, 1,
ylab = expression(paste("Núcleo de ", f[x](x))))

curve(fy(y = x, x = 0.5), xlab = expression(y), 0, 1,
ylab = expression(paste("Núcleo de ", f[y](y))))

```





1) Gerador de amostras

Utiliza-se o algoritmo de Metropolis-Hastings. Para gerar (X,Y) , adota-se uma distribuição proposta uniforme bivariada com parâmetros $(0,1)$.

```
set.seed(6518)
#constantes
M <- 3000
descarte <- 100
espac <- 10
nsim <- descarte + M*espac
# Valores iniciais
x <- y <- c()
x[1] <- 0.5
y[1] <- 0.5
# Contadores de aceitações
nac <- 0

# Simulação
for (j in 2:nsim) {
  # candidato x,y
  xc <- runif(1)
  yc <- runif(1)
  # probabilidade alfa
  alfa <- min(1, fxy(xc,yc)/fxy(x[j-1],y[j-1]))
  # geração x,y
  if (runif(1) <= alfa) {
    x[j] <- xc
    y[j] <- yc
    nac <- nac + 1
  } else {
    x[j] <- x[j - 1]
    y[j] <- y[j - 1]
  }
}
```

Ajuste de espaçamento e descarte

```
x <- x[-(1:descarte)]
y <- y[-(1:descarte)]

xa <- ya <- c()
j= 1
for (i in 1:(length(x)/espac)) {
```

```

    xa[i]= x[j]
    ya[i]= y[j]
    j= j + espac
}
x <- xa
y <- ya

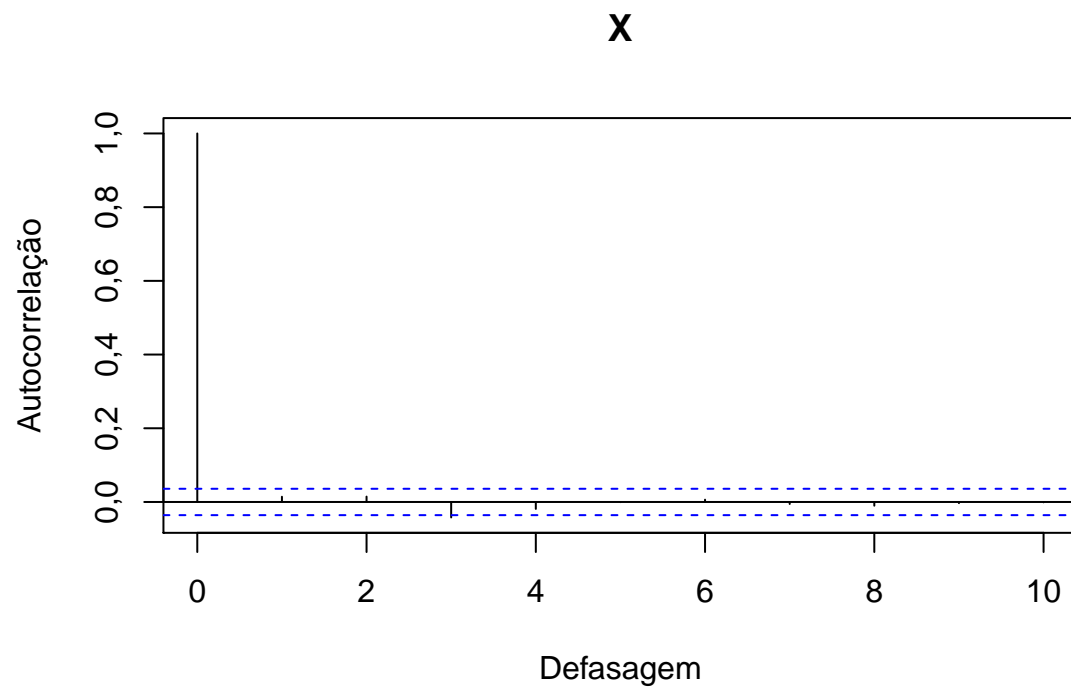
```

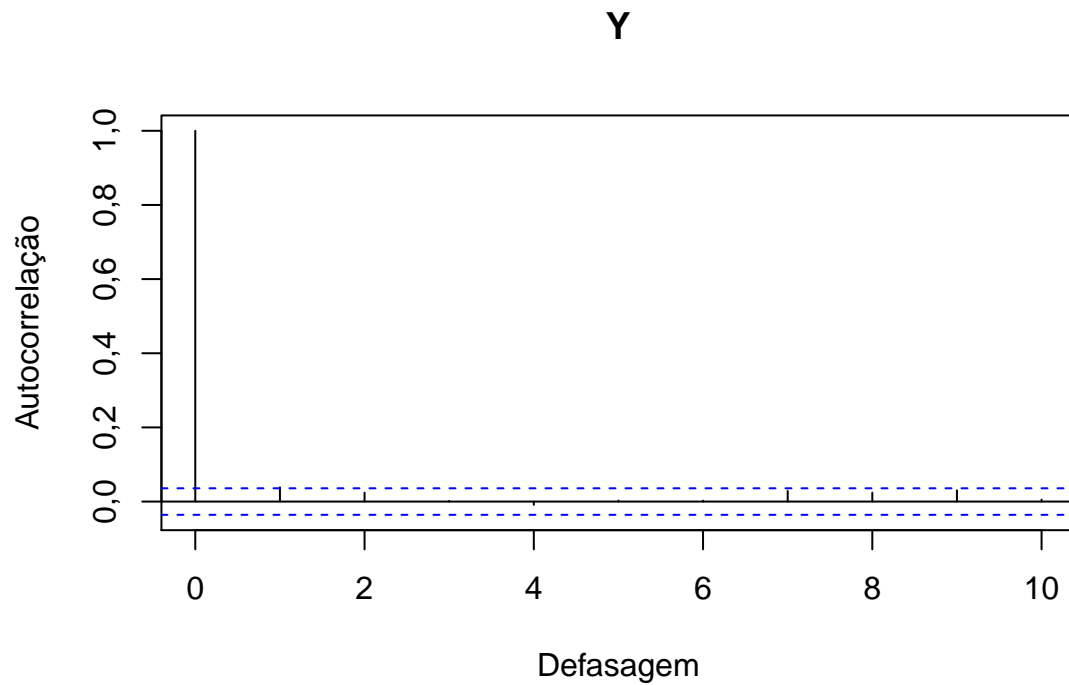
Gráficos de defasagem

```

acf(x, lag.max = 10, main = "X", xlab = "Defasagem", ylab = "Autocorrelação")
acf(y, lag.max = 10, main = "Y", xlab = "Defasagem", ylab = "Autocorrelação")

```





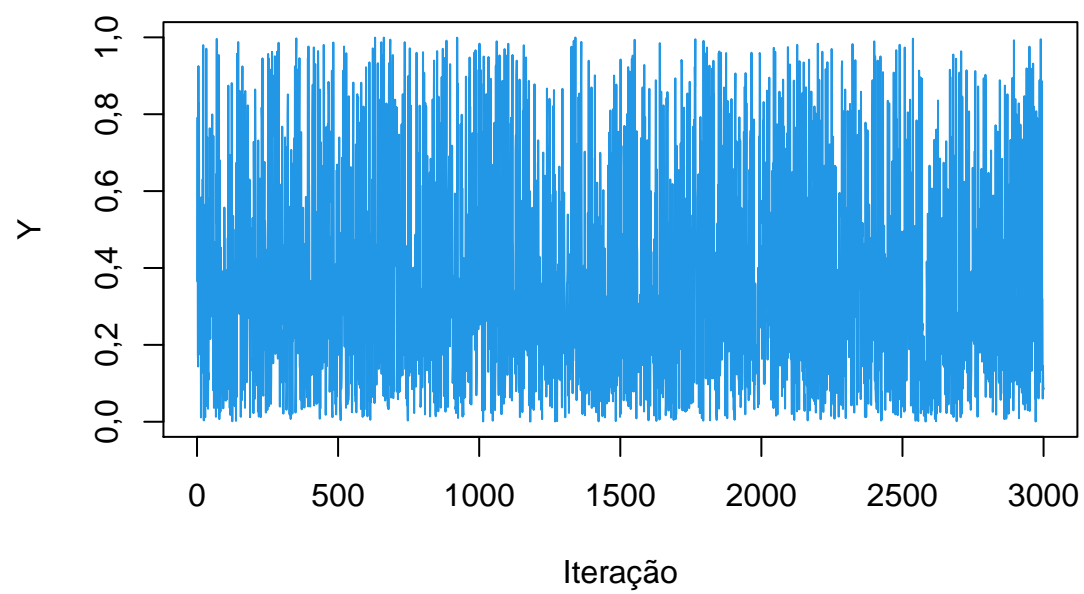
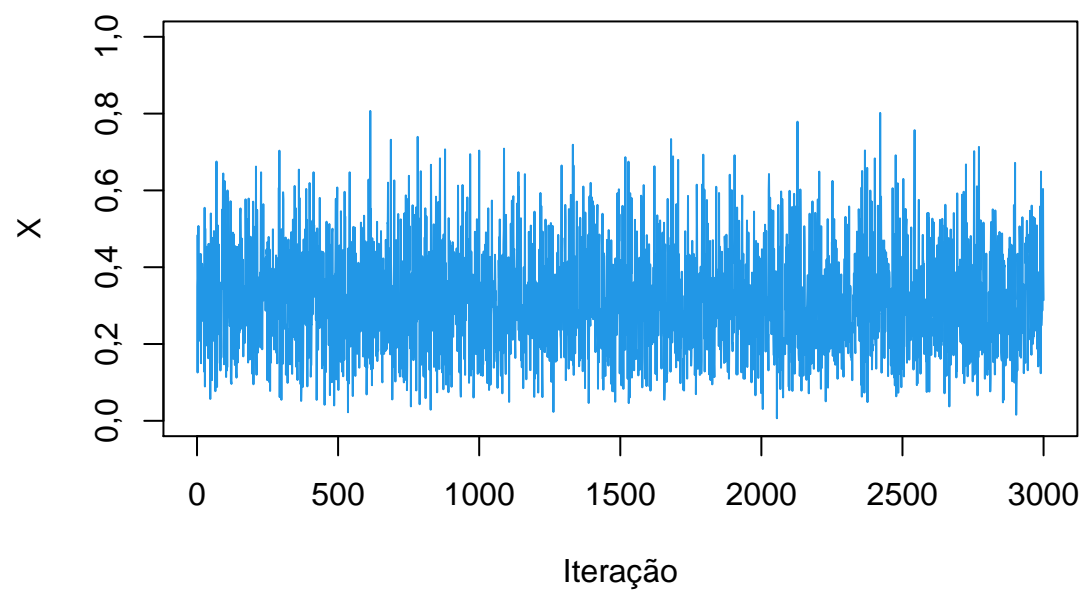
2) Estimativa da taxa de aceitação

```
cat("\n Taxa de aceitação (%):", round(nac / (nsim - 1) * 100, 1))
```

Taxa de aceitação (%): 37,5

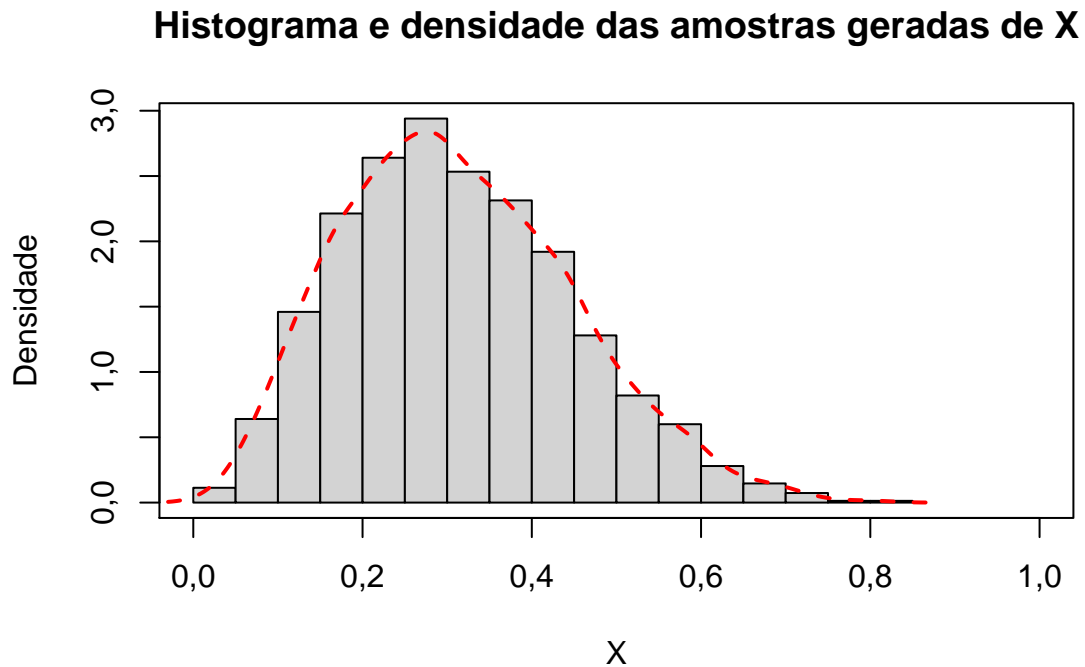
3) Gráficos das cadeias geradas

```
plot(x, type = "l", xlab = "Iteração", ylab = expression(X), col = 4, ylim = c(0,1))
plot(y, type = "l", xlab = "Iteração", ylab = expression(Y), col = 4)
```

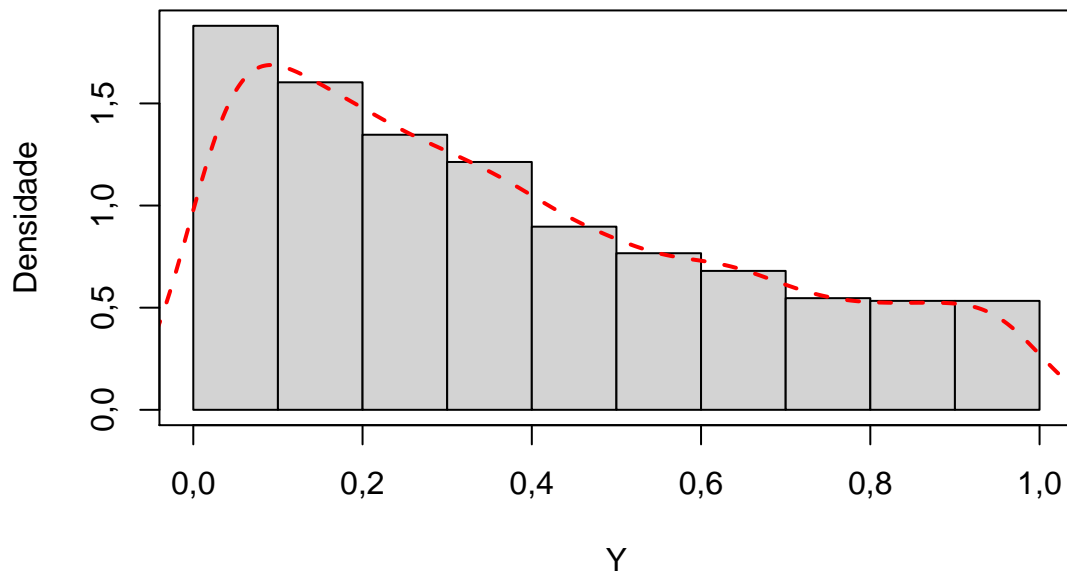


4) Histogramas e densidades das amostras geradas

```
hist(x, freq = FALSE, main = "Histograma e densidade das amostras geradas de X",  
     xlab = expression(X), ylab = "Densidade", xlim = c(0,1))  
lines(density(x), col = "red", lty = 2, lwd = 2)  
box()  
  
hist(y, freq = FALSE, main = "Histograma e densidade das amostras geradas de Y",  
     xlab = expression(Y), ylab = "Densidade")  
lines(density(y), col = "red", lty = 2, lwd = 2)  
box()
```



Histograma e densidade das amostras geradas de Y



Pode-se observar que os histogramas se assemelham aos gráficos das distribuições dos núcleos das condicionais completas, o que é esperado.

5) Medidas resumo das distribuições de X e Y

```
# Separador decimal: "."
# Necessário para o r markdown imprimir os valores da tabela corretamente
options(OutDec = ".")

#função para estimar o ponto máximo do gráfico de densidade, estimando a moda
moda_estimada <- function(x) {
  hx <- hist(x,breaks = 50,freq = F)
  return(hx$mids[which.max(hx$density)])
}
```

Medida	X	Y
Média	0.3130153	0.3714769
Mediana	0.2997182	0.3104781
Moda Estimada	0.29	0.03
Desvio padrão	0.1363338	0.2770116