INF1036 - Probabilidade Computacional – 2ª. Lista – 2023.1

Professora: Ana Carolina Letichevsky

1) Suponha que a fábrica de um certo equipamento fez uma pesquisa e classificou as 4 principais causas de defeito no equipamento, quando o mesmo possui entre dois e três anos de uso como C1, C2, C3 e C4. Considere ainda que:

- sem a manutenção preventiva no período recomendado pela fábrica, o equipamento pode apresentar os defeitos C1, C2, C3 e C4, com as probabilidades, respectivamente, de 4%, 4%, 6% e 6%;
- realizar a manutenção preventiva no período indicado pela fábrica diminui o risco dos defeitos C3 e C4 em 50%, mas não afeta o risco de C1 e C2;
- as eventuais ocorrências de defeito (com ou sem manutenção) são independentes;
- 40% dos equipamentos realizam a manutenção preventiva no prazo certo.

Um equipamento é selecionado aleatoriamente. Calcule, usando simulação, a probabilidade de que entre dois e três anos:

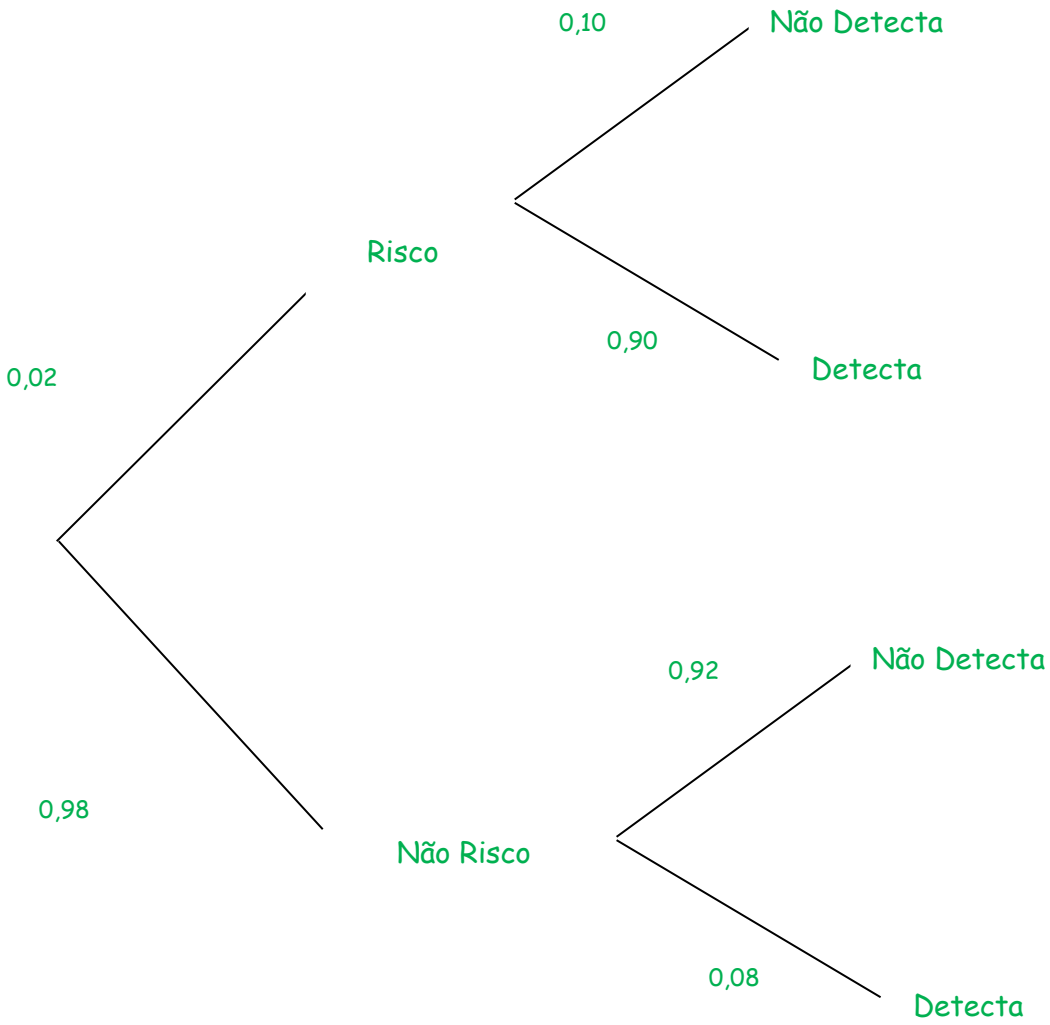
a) o equipamento:
   - apresente defeito;
   - apresente algum defeito se não for feita manutenção preventiva;
   - apresente algum defeito se for feita a manutenção preventiva.

b) tenha sido feita manutenção preventiva dado que o equipamento apresentou apenas o defeito:
   - C 1;
   - C 3.

```
nsamples <- 1000000
contsemdefeito <- 0
contcomdefeito <- 0
contcommanutencao <- 0
contsemmanutencao <- 0
contcommanutencaosemdefeito <- 0
contcommanutencaocomdefeito <- 0
contsemmanutencaosemdefeito <- 0
contsemmanutencaocomdefeito <- 0
contcommanutencaocomdefeitoC1 <- 0
contsemmanutencaocomdefeitoC1 <- 0
contcommanutencaocomdefeitoC3 <- 0
contsemmanutencaocomdefeitoC3 <- 0

for (i in 1:nsamples) {
  efetuamanutencao <- sample(c(T, F), 1, prob = c(0.4, 0.6))
  if (efetuamanutencao == T) {
    contcommanutencao <- contcommanutencao + 1
    apresentoudefeitoC1 <- sample(c("C1", "SD"), 1, prob = c(0.04, 0.96))
    apresentoudefeitoC2 <- sample(c("C2", "SD"), 1, prob = c(0.04, 0.96))
    apresentoudefeitoC3 <- sample(c("C3", "SD"), 1, prob = c(0.03, 0.97))
    apresentoudefeitoC4 <- sample(c("C4", "SD"), 1, prob = c(0.03, 0.97))

    if ((apresentoudefeitoC1 == "C1") & (apresentoudefeitoC2 == "SD") &
(apresentoudefeitoC3 == "SD") & (apresentoudefeitoC4 == "SD")) {
      contcommanutencaocomdefeitoC1 <- contcommanutencaocomdefeitoC1 + 1
```

```
    }
    if ((apresentoudefeitoC1 == "SD") & (apresentoudefeitoC2 == "SD") &
(apresentoudefeitoC3 == "C3") & (apresentoudefeitoC4 == "SD")) {
      contcommanutencaocomdefeitoC3 <- contcommanutencaocomdefeitoC3 + 1
    }

    if ((apresentoudefeitoC1 == "SD") && (apresentoudefeitoC2 == "SD") &&
(apresentoudefeitoC3 == "SD") && (apresentoudefeitoC4 == "SD")) {
      contcommanutencaosemdefeito <- contcommanutencaosemdefeito + 1
    } else {
      contcommanutencaocomdefeito <- contcommanutencaocomdefeito + 1
    }

  } else {
    contsemmanutencao <- contsemmanutencao + 1
    apresentoudefeitoC1 <- sample(c("C1", "SD"), 1, prob = c(0.04, 0.96))
    apresentoudefeitoC2 <- sample(c("C2", "SD"), 1, prob = c(0.04, 0.96))
    apresentoudefeitoC3 <- sample(c("C3", "SD"), 1, prob = c(0.06, 0.94))
    apresentoudefeitoC4 <- sample(c("C4", "SD"), 1, prob = c(0.06, 0.94))

    if ((apresentoudefeitoC1 == "C1") & (apresentoudefeitoC2 == "SD") &
(apresentoudefeitoC3 == "SD") & (apresentoudefeitoC4 == "SD")) {
      contsemmanutencaocomdefeitoC1 <- contsemmanutencaocomdefeitoC1 + 1
    }
    if ((apresentoudefeitoC1 == "SD") & (apresentoudefeitoC2 == "SD") &
(apresentoudefeitoC3 == "C3") & (apresentoudefeitoC4 == "SD")) {
      contsemmanutencaocomdefeitoC3 <- contsemmanutencaocomdefeitoC3 + 1
    }

    if ((apresentoudefeitoC1 == "SD") && (apresentoudefeitoC2 == "SD") &&
(apresentoudefeitoC3 == "SD") && (apresentoudefeitoC4 == "SD")) {
      contsemmanutencaosemdefeito <- contsemmanutencaosemdefeito + 1
    } else {
      contsemmanutencaocomdefeito <- contsemmanutencaocomdefeito + 1
    }
  }
}

contsemdefeito <- contcommanutencaosemdefeito + contsemmanutencaosemdefeito
contcomdefeito <- contcommanutencaocomdefeito + contsemmanutencaocomdefeito

#ai
print(contcomdefeito/nsamples)

#aii
print(contsemmanutencaocomdefeito/contsemmanutencao)

#aiii
print(contcommanutencaocomdefeito/contcommanutencao)

#bi
print(contcommanutencaocomdefeitoC1/(contcommanutencaocomdefeitoC1 +
contsemmanutencaocomdefeitoC1))

#bii
print(contcommanutencaocomdefeitoC3/(contcommanutencaocomdefeitoC3 +
contsemmanutencaocomdefeitoC3))
```

2) Os detectores de mentira já foram utilizados durante guerras para revelar riscos de segurança. Como é sabido, os detectores de mentira não são infalíveis. Suponhamos que haja uma probabilidade de 0,10 de um detector de mentiras não detectar uma pessoa que represente realmente um risco de segurança e uma probabilidade de 0,08 de o detector classificar incorretamente uma pessoa que não represente um risco. Se 2% das pessoas que são submetidas ao teste constituem efetivamente risco de segurança, calcule a probabilidade de que:



a) uma pessoa classificada pelo detector como constituindo um risco de segurança constitua realmente um risco de segurança;

$$P(Risco/Detecta) = \frac{0{,}02 \times 0{,}90}{(0{,}02 \times 0{,}90) + (0{,}98 \times 0{,}08)} = \frac{0{,}018}{0{,}0966} = 0{,}1867$$

b) uma pessoa liberada pelo detector realmente não constitua um risco de segurança.

$$P(\text{Não Risco}/\text{Não Detecta}) = \frac{0{,}98 \times 0{,}92}{(0{,}98 \times 0{,}92) + (0{,}02 \times 0{,}10)} = \frac{0{,}9016}{0{,}9036} = 0{,}9978$$

3) Uma empresa encomenda 20 placas de vídeo de um fornecedor novo para montagem nos seus micros. A probabilidade de defeito numa placa é, segundo o fornecedor, de 10%. A empresa toma uma amostra de 10 micros. Qual é a probabilidade de 4 deles apresentarem uma placa de vídeo com defeito se:

X = nº de peças defeituosas.

a) a amostra é com reposição?

$$X \sim Binomial(10; \quad 0{,}1)$$

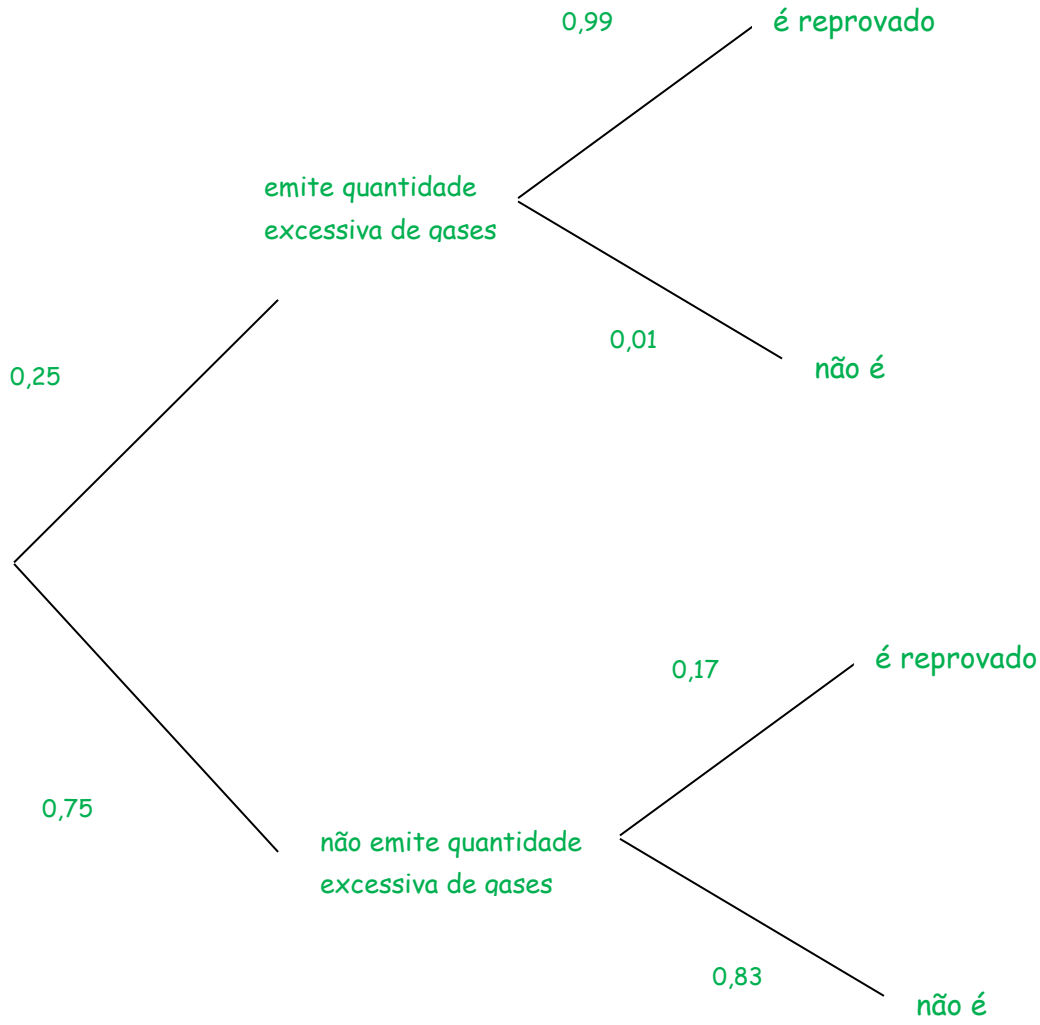
$$P(X = 4) = C_{10}^{4} \times 0,1^{4} \times 0,9^{6} = 0{,}01116$$

b) a amostra é sem reposição?

$$X \sim Hipergeométrica$$

$$P(X = 4) = 0$$

4) Num certo estado onde os automóveis devem ser testados quanto à emissão de gases poluentes, 25% de todos os automóveis emitem quantidade excessivas de gases poluentes. Ao serem testados, 99% de todos os automóveis que emitem quantidades excessivas de gases poluentes são reprovados, mas 17% dos que não emitem quantidades excessivas de gases poluentes também são reprovados. Qual é probabilidade de um automóvel que é reprovado no teste efetivamente emitir uma quantidade excessiva de gases poluentes?



A = automóvel ser reprovado no teste;

B = automóvel efetivamente emitir uma quantidade excessiva de gases poluentes.

$$P(B/A) = \frac{P(A \cap B)}{P(A)}$$

$$P(A \cap B) = 0{,}25 \times 0{,}99 = 0{,}248$$

$$P(A) = (0{,}25 \times 0{,}99) + (0{,}75 \times 0{,}17) = 0{,}375$$

$$P(B/A) = 0{,}66$$

5) Suponha que há 3 moedas, cada uma de uma cor diferente, mas todas com probabilidades $3/5$ de obter coroa e $2/5$ de obter cara.

Considere um experimento que consiste em lançar, em sequência as moedas.

Considere também dois eventos:

A = "obter uma coroa e uma cara nos dois primeiros lançamentos, em qualquer ordem", e

B = "obter duas coroas nos dois últimos lançamentos".

Considerando o que foi exposto, resolva os itens abaixo:

a) Utilizando simulação calcule $P(A)$ e $P(B)$.

```
nsamples <- 10000
cont.A <- 0
cont.B <- 0
cont.AeB <- 0
for (i in 1:nsamples) {
  A.verdadeiro <- FALSE
  B.verdadeiro <- FALSE
  face1 <- sample(c("CA","CO"), 1, prob = c(2/5, 3/5), replace = TRUE)
  face2 <- sample(c("CA","CO"), 1, prob = c(2/5, 3/5), replace = TRUE)
  face3 <- sample(c("CA","CO"), 1, prob = c(2/5, 3/5), replace = TRUE)
  if (((face1 == "CA") & (face2 == "CO")) | ((face1 == "CO") & (face2 == "CA"))) {
    cont.A <- cont.A + 1
    A.verdadeiro <- TRUE
  }
  if ((face2 == "CO") & (face3 == "CO")) {
    cont.B <- cont.B + 1
    B.verdadeiro <- TRUE
  }
  if ((A.verdadeiro) & (B.verdadeiro)) {
    cont.AeB <- cont.AeB + 1
  }
}

#a)

PA <- cont.A / nsamples
print(PA)

PB <- cont.B / nsamples
print(PB)
```

b) A e B são eventos independentes?

```
PAeB <- cont.AeB / nsamples
print(PAeB)

PAxPB <- PA * PB
print(PAxPB)
# Como PAeB é diferente de PAxPB então os eventos não são independentes.
```

6) Considerando $g(x) = x^2 + 3x$ uma função, calcule o valor aproximado de φ, onde:

$$\varphi = \int_0^1 g(x)dx$$

Observação:

Considerando U uma variável aleatória uniformemente distribuída no intervalo $[0, 1]$, então:

$$\varphi = \int_0^1 g(x)dx = E[g(U)]$$

Se $U_1, U_2, \dots, U_n$ são variáveis aleatórias independentes e uniformemente distribuídas em $[0, 1]$, tem-se que as variáveis aleatórias $g(U_1), g(U_2), \dots, g(U_n)$ são variáveis aleatórias independentes com média φ. Portanto, pela Lei Forte dos Grandes Números (Lei Forte de Kolmogorov), segue que, com probabilidade 1,

$$\sum_{i=1}^{n} \frac{g(U_i)}{n} \to E[g(U)] = \varphi$$

onde $n \to \infty$.

Para realizar o cálculo, deve ser criada uma função chamada **aproximador** que deverá retornar o valor aproximado de φ. Implemente a função **aproximador**, em R e em Python, usando como gerador de números pseudoaleatórios o algoritmo:

a) LGC.

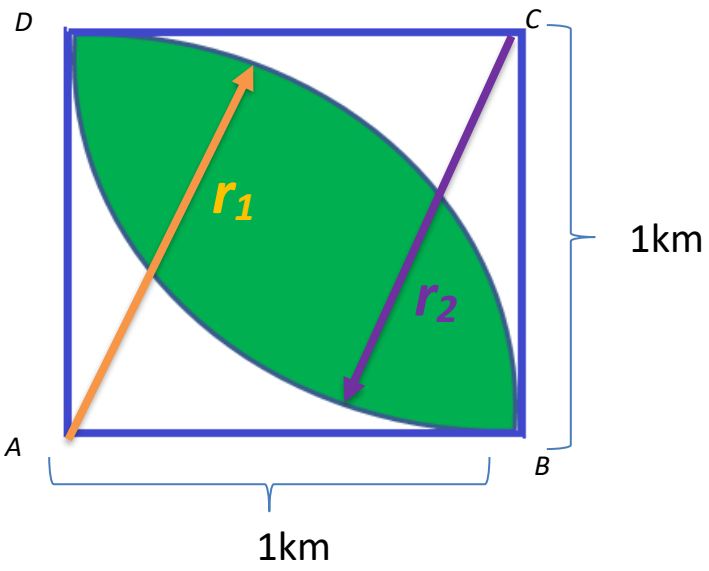
```
#R
#a)
LCG <- function (seed, a, c, M, nsamples) {
  x <- seed
  u <- NULL
  for (i in 1:nsamples) {
    nx <- (a * x + c) %% M
    u <- c(u, as.double(nx) / as.double(M))
    x <- nx
  }
  return (u)
}

aproximador <- function(U) {
  retorno <- NULL
  n <- length(U)
  vetor = NULL
  for (i in 1:n) {
    gx <- U[i]^2 + 3*U[i] #cálculo da g(x)
    vetor <- c(vetor, gx)
  }
  retorno <- mean(vetor)
  return (retorno)
}
```

```
a <- 39373
c <- 0
M <- 2147483647
nsamples <- 1000
U = LCG(3, a, c, M, nsamples)
valor.aproximado <- aproximador(U)
print(valor.aproximado)

#Python
#a)
import numpy as np
def LCG(seed, a, c, M, nsamples):
    x = seed
    u = []
    u.append(x)
    for i in range(nsamples - 1):
        nx = (a * x + c) % M
        u.append(float(nx) / float(M))
        x  = nx
    return (u)

def aproximador (U):
  retorno = 0
  n = len(U)
  vetor = []
  for i in range(0, n):
    gx = U[i]**2 + 3*U[i] #cálculo da g(x)
    vetor.append(gx)
  retorno = np.mean(vetor)
  return (retorno)

a = 39373
c = 0
M = 2147483647
nsamples = 1000

U = LCG(3, a, c, M, nsamples)
valoraproximado = aproximador(U)
print(valoraproximado)
```

b) Mersenne Twister.

```
#R
#b)
aproximador <- function(U) {
  retorno <- NULL
  n <- length(U)
  vetor <- NULL
  for (i in 1:n) {
    gx <- U[i]^2 + 3*U[i] #cálculo da g(x)
    vetor <- c(vetor, gx)
  }
  retorno <- mean(vetor)
  return (retorno)
}

nsamples <- 1000
U <- runif(nsamples)

valor.aproximado <- aproximador(U)
print(valor.aproximado)

#Python
#b)
import numpy as np
def aproximador (U):
  retorno = 0
  n = len(U)
  vetor = []
  for i in range(0, n):
    gx = U[i]**2 + 3*U[i] #cálculo da g(x)
    vetor.append(gx)
  retorno = np.mean(vetor)
  return (retorno)

nsamples = 1000
U = np.random.sample(nsamples)
valoraproximado = aproximador(U)
print(valoraproximado)
```

7) Na figura abaixo, a área verde representa a área cultivada de uma fazenda que está inserida em uma região quadrada de lado 1km e vértices A, B, C e D.



Na figura, temos que, r_1 e r_2 representam os raios de duas circunferências de raio 1km. O centro da circunferência de raio r_1 coincide com o vértice A da região quadrada, e o centro da circunferência de raio r_2 coincide com o vértice C da região quadrada.

Sabendo que, para adubar a área o fazendeiro precisa comprar 100kg de adubo por quilômetro quadrado por mês, estime, por simulação, qual a quantidade de adubo ele deverá comprar em 12 meses?

```
nsamples <- 10000
cont.dentro.area.cultivada <- 0

for (i in 1:nsamples) {
  xy.r1 <- runif(2)
  xy.r1.soma.quadrado <- sum(xy.r1^2)
  xy.r2 <- 1 - xy.r1
  xy.r2.soma.quadrado <- sum(xy.r2^2)
  if ((sqrt(xy.r1.soma.quadrado) < 1) & (sqrt(xy.r2.soma.quadrado) < 1))
    cont.dentro.area.cultivada <- cont.dentro.area.cultivada + 1
}

p.dentro.area.cultivada <- cont.dentro.area.cultivada / nsamples
area.total <- 1
area.cultivada = p.dentro.area.cultivada * area.total

adubo.por.quilometro <- 100
numero.meses <- 12
quantidade.adubo <- area.cultivada * adubo.por.quilometro * numero.meses

print(paste0("Quantidade (kg) = ", quantidade.adubo))
```