Aula Especial: Cadeias de Markov e Processos de Poisson

Nesta aula, vamos trabalhar com o auxílio do computador diversas questões das listas de exercícios. Durante a aula, serão explicados os problemas e esperamos que os exemplos ajudem a resolução dos desafios propostos. Focalizarmos os significados e não os detalhes técnicos da linguagem `R`. Esperamos que as simulações ajudem a evidenciar resultados teóricos vistos em sala de aula. Serão formadas duplas e os cinco desafios propostos deverão ter seus códigos em linguagem `R` entregues com nomes e números USP até às 23:59 do dia 30/05/2018 para o e-mail do monitor (`moura_caio@yahoo.com.br`). Caso algum exercício não tenha sido concluído, explique a ideia da resolução e possível motivo do código não ter funcionado ou do resultado não ter sido alcançado.
Obs.: Quando um pedaço do enunciado original de um problema da lista foi alterado, esse aparece em itálico.

1. **Lista 1, Questão 7**

   Suponha que cada item produzido por uma fábrica é classificado como defeituoso ou perfeito. Se um item é defeituoso ou não, depende da qualidade do item previamente produzido. Suponha que um item defeituoso é seguido por outro item defeituoso com probabilidade 2/3, enquanto que um item perfeito é seguido de outro item perfeito com probabilidade 3/4. Suponha que no instante zero, um item perfeito é produzido. Determine a probabilidade de que o terceiro item produzido é defeituoso. *Encontre também a distribuição estacionária para esse problema.*

Solution: Aqui podemos montar a matriz

$$M = \begin{array}{c c} & \begin{array}{cc} D & D^C \end{array} \\ \begin{array}{c} D \\ D^C \end{array} & \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{4} & \frac{3}{4} \end{pmatrix} \end{array},$$

sendo que a primeira linha representa as probabilidades de partir de um estado defeituoso D e a primeira coluna representa as probabilidades de chegar a um estado defeituoso D em uma unidade de tempo. O enunciado pede para determinar a probabilidade em três unidades de tempo sendo que não sabemos quais são os estados intermediários X_1 e X_2. Em notação matemática, queremos determinar

$$\mathbb{P}(X_3 = D | X_0 = D).$$

Uma forma de simplificar os cálculos é obter as potências da matriz de transição. Partindo de um estado i, as probabilidades de chegar a um estado $i+n$, sendo que não sabemos quais são os estados intermediarios, podem ser calculadas como

$$M^n = \begin{array}{c c} & \begin{array}{cc} D & D^C \end{array} \\ \begin{array}{c} D \\ D^C \end{array} & \begin{pmatrix} \mathbb{P}(X_{n+i} = D | X_i = D) & \mathbb{P}(X_{n+i} = D^C | X_i = D) \\ \mathbb{P}(X_{n+i} = D | X_i = D^C) & \mathbb{P}(X_{n+i} = D^C | X_i = D^C) \end{pmatrix} \end{array},$$

Outra forma de encontrar a resposta é através autovalores e autovetores. Temos que este processo é irredutível, recorrente, não-nulo e aperiódico, então há distribuição limitante $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_n)$ com $\pi_i > 0$. E a distribuição limitante, quando existe, é idêntica à distribuição estacionária.

Utilizamos o seguinte código em `R` para multiplicar matrizes:

```
potM <- function(M, n){
if (n == 1)  return (M)
if (n == 2)  return (M %*% M)
if (n > 2) return ( M %*% potM(M, n-1))
}
```

Assim, a matriz que procuramos é obtida com o seguinte código:

```
M <- matrix(c(2/3,1/3 ,1/4,3/4), byrow = T, ncol = 2)
potM(M, 3)
```

Agora vamos encontrar a distribuição limitante. Uma forma de fazê-lo é elevando a matriz de transição a potências altas até que se verifique a convergência da matriz. Com o comando `potM(M, 2000)`, por exemplo. Uma maneira mais sofisticada é através de autovalores e autovetores. Veja que

$$\boldsymbol{\pi M} = \boldsymbol{\pi} \Rightarrow \boldsymbol{M}^\intercal \boldsymbol{\pi}^\intercal = \boldsymbol{\pi}^\intercal, \text{ sendo } \boldsymbol{\pi} \text{ um vetor-linha da distribuição limitante.}$$

Logo, podemos calcular os autovalores e autovetores da transformação $\boldsymbol{M}^\intercal$ para obter uma aproximação de $\lim_{k\to\infty} (\boldsymbol{M^k})^\intercal$ e, portanto, de $\lim_{k\to\infty} \boldsymbol{M^k}$. Com o comando `eigen(M)` obtemos os autovalores e os autovetores:

```
auto <- eigen(t(M))
autova <- auto$values
autove <- auto$vectors

P <- as.matrix(autove)
Dk <- diag(autova^2000)
Mkt <- P %*% Dk %*% solve(P) # A funcao 'solve' inverte a matriz
t(Mkt)
```

Exercício em sala:

Use a função `potM` e a técnica com autovalores e autovetores para calcular $\mathbb{P}(X_3 = 2|X_0 = 0)$ do processo da Questão 5 da Lista 1. Para ajudar, informamos a matriz:

`N <- matrix(c(.3,.3,.4, .2,.6,.2, 0,0,1), byrow = T, ncol = 3)` .

2. **Lista 1, Questão 9**

   Os físicos Paul e Tatyana Ehrenfest consideram um modelo conceitual para o movimento de moléculas no qualM moléculas estavam distribuídas entre 2 urnas. Em cada instante de tempo uma das moléculas era escolhida aleatoriamente, removida de sua urna e colocada na outra. Se X_n representa o número de moléculas na primeira urna imediatamente após a n-ésima mudança então $(X_n)_{n\geq 0}$ é uma cadeia de Markov. *Calcule as potências $n = 37, 38, 39, 40$ da matriz de transição. Quais são as propriedades que podemos observar neste processo? Encontre os autovalores da transposta da matriz de transição e argumente se este processo possui distribuição estacionária. Caso afirmativo, apresente a distribuição estacionária e justifique se esta pode ser obtida por meio de potências da matriz de transição. Argumente se os resultados obtidos com as potências têm relação com os autovalores encontrados.*

Solution:

Nesta questão oferecemos o código que gera uma matriz de transição para o modelo da urna de Ehrenfest.

```r
# Modelo de Ehrenfest com n+1 espacos
greatest.estate <- 4

# Construindo a matrix de transicao
matriz <- function(greatest.estate){
  M <- matrix(0, nrow = greatest.estate+1, ncol = greatest.estate+1)
  M[1, 2] <- 1
  M[greatest.estate+1, greatest.estate] <- 1
  for(j in 2:greatest.estate){
    M[j, j-1] <- (j-1)/greatest.estate
    M[j, j+1] <- (greatest.estate - (j-1))/greatest.estate
  } return(M) }
```

3. **Lista 1, Questão 12**

Considere as seguintes cadeias de Markov:

(a) O passeio aleatório em $\mathbb{Z}$;

(b) O passeio aleatório no círculo com estados 0, 1, 2, 3, 4 e 5;

(c) A cadeia consideradas nos exercícios na Questão 4 da Lista 1.

Construa um código que gere uma simulação para cada um dos itens de forma que se faça uso de realizações de uma variável aleatória uniforme $U \sim U(0,1)$. *Para o item a), considere também o parâmetro* $p = 0.50$, $p = 0.40$ *e* $p = 0.60$ *descrevendo qual o comportamento do processo simulado para cada valor.*

Solution: Podemos pensar na variável aleatória U como uma espécie de dado de infinitas faces. Dependendo da realização da variável aleatória U decidimos qual é o próximo estado X_n dado que conhecemos o estado X_{n-1}.

Propomos uma solução para o item (a) em que simulamos até o tempo 60. Assim,

```r
tempo.total <- 61
est.inicial <- 0
estados <- numeric()
p <- 0.50
estados[1] <- est.inicial
for(i in 2:tempo.total){
u <- runif(n = 1, min = 0, max = 1)
estados[i] <- ifelse(test = u <= p, yes = estados[i-1] + 1, no = estados[i-1] - 1)
}
estados

# O codigo abaixo cria uma representacao grafica para o processo
plot(c(0, tempo.total), c(-tempo.total*(abs(p-.5)+0.4), tempo.total*(abs(p-.5)+0.4)), type = "n",
 main = "Passeio aleatorio", xlab = "Tempo", ylab = "Estado")
grid(nx = NULL, ny = NULL, col = "lightgray", lty = "dotted", lwd = par("lwd"), equilogs = TRUE)
lines(0:(tempo.total-1), estados, col="indianred")
```

Propomos também uma solução o item (c) até o tempo 60. Trata-se de uma solução que faz uso dos índices da matriz de transição para decidir as probabilidade a cada tempo n.

```r
M <- matrix(c(.1,.1,.8, .2,.2,.6, .3,.3,.4), byrow = T, ncol = 3)
# Calcular a distribuicao limitante nos da uma intuicao do que se espera da simulacao
potM(M,2000)

tempo.total <- 61
est.inicial <- 0
estados <- numeric()
estados[1] <- est.inicial
for(i in 2:tempo.total){
    u <- runif(n = 1, min = 0, max = 1)
    if(u <= M[estados[i-1] + 1, 1]){
        estados[i] <- 0
    }
    if(M[estados[i-1] + 1, 1] < u && u<= M[estados[i-1] + 1, 1] + M[estados[i-1] + 1, 2]){
        estados[i] <- 1
    }
    if(M[estados[i-1] + 1, 1] + M[estados[i-1] + 1, 2] < u && u <=
    M[estados[i-1] + 1, 1] + M[estados[i-1] + 1, 2]+ M[estados[i-1] + 1, 3]){
        estados[i] <- 2
    }
}
estados
table(estados)

# O codigo abaixo cria uma representacao grafica para o processo
plot(c(0, tempo.total), c(0,2), type = "n", main = "", xlab = "Tempo", ylab = "Estado")
grid(nx = NULL, ny = NULL, col = "lightgray", lty = "dotted", lwd = par("lwd"), equilogs = TRUE)
lines(0:(tempo.total-1), estados, col="indianred")
```

Com base nestes exemplos, construa a simulação para o item (b) e sua representacao gráfica. Se achar útil, utilize a operação **%%** (resto de divisão inteira).

4. **Inspirado no Paradoxo da Partida do livro 'Introduction to Stochastic Processes With R', de Dobrow (2016, p.255)**

Suponha que Catarina sempre chega no ponto de ônibus às 5h45 da manhã. A empresa de ônibus libera seus ônibus de acordo com um processo de Poisson com taxa de 5,5 ônibus por hora. O processo se inicia diariamente às 5h. Qual é o tempo médio de espera de Catarina durante 1000 dias?

Solution: Geramos os tempos de chegada de um processo de Poisson até às 12h. Por simplicidade 5h equivale ao tempo 0 e 12h equivale ao tempo 7. Catarina chega, portanto, no tempo 0,75.

```r
# Inicio um vetor de tamanho variavel
tempos <- numeric(0)
tempo_da_catarina <- 0.75
i <- 1
# Tempo do inicio do processo eh zero
t <- 0
while(t < 7){
  tempos[i] <- t
  # A funcao rexp gera o tempo de chega de um onibus
  t <- t + rexp(1, rate = 5.5)
  i <- i + 1
}
# Obtenho todos tempos de chegada
tempos
total_onibus <- length(tempos) - 1

# Para saber qual o tempo do onibus que Catarina pegou, pergunto quais os indices do vetor tempo
# cujo valor eh acima de 0.75 e escolho o menor indice, pois o vetor esta ordenado.
which(tempos > tempo_da_catarina)
indice_onibus_catarina <- min(which(tempos > tempo_da_catarina))
espera <- tempos[indice_onibus_catarina] - tempo_da_catarina
espera
```

Agora que descobrimos o tempo de espera para 1 dia, como fazer para 1000 dias?

Exercício em sala:

Suponha agora que Catarina pegue seu ônibus às 8h30. Agora queremos saber o tamanho do intervalo de tempo entre o ônibus que Catarina pegou e o ônibus anterior a este. Simule esse evento 1000 vezes e informe a média dos intervalos obtidos. Compare essa média com o dobro de 1/5,5.

5. **Inspirado no livro 'Introduction to Stochastic Processes With R', de Dobrow (2016)**

Segundo Dobrow (2016, p.250), em processo de poisson espacial, "Dado um conjunto $A \subseteq \mathbb{R}^d (d \geq 1)$, então condicionando na existência de n pontos em A,, os locais dos pontos são uniformementes distribuídos em A. Por isso, um processo de Poisson espacial é às vezes chamado um modelo de *aleatoriedade espacial completa*." Mostramos como gerar através da linguagem `R`, uma realização do processo de Poisson bidimensional.

```
lambda <- 7 # por quilometro quadrado
area <- 36  # trata-se de um quadrado de lado 5
N <- rpois(1, lambda * area) # fixa o numero de pontos total
xcoord <- runif(N, 0, 6)
ycoord <- runif(N, 0, 6)
plot(xcoord, ycoord, cex=0.7, ylab = "", xlab = "")
```

Exercício em sala:

Na natureza, diversas plantas possuem sementes que podem voar para lugares distantes. Por simulação, geramos coordenadas de queda de sementes de uma espécie determinada em uma área e as armazenamos nos arquivos `base0_coord_x.csv` e `base0_coord_y.csv`. Para saber qual base escolher, use o seguinte critério: dentre os números USP dos integrantes da sua dupla, escolha o menor número e observe sua paridade. Escolha as coordenadas x e y da base 02 se for par. Caso contrário, escolha as coordenadas da base 01. Forneça pelo menos dois indícios, os melhores possíveis, de que sua base possue pontos distribuídos como um processo de Poisson ou não. Ambas possuem coordenadas de pontos contidas num quadrado de lado 5.

Coloque seus arquivos no seu `working directory`. Um exemplo de como estabelecer qual seu `working directory` (com o comando `setwd`) e importar os dados das coordenadas é dado abaixo:

```
setwd("~/Dropbox/2018") # serviu para a pasta '2018' do dropbox
x <- as.matrix(unname(  read.csv("base02coord_x.csv", header = TRUE) ) )
y <- as.matrix(unname(  read.csv("base02coord_y.csv", header = TRUE) ) )
```