Pacotes markvochain e diagram

PROCESSOS ESTOCÁSTICOS

UNB

ALINE
ANANDA
JULIANA
KHÉZIA

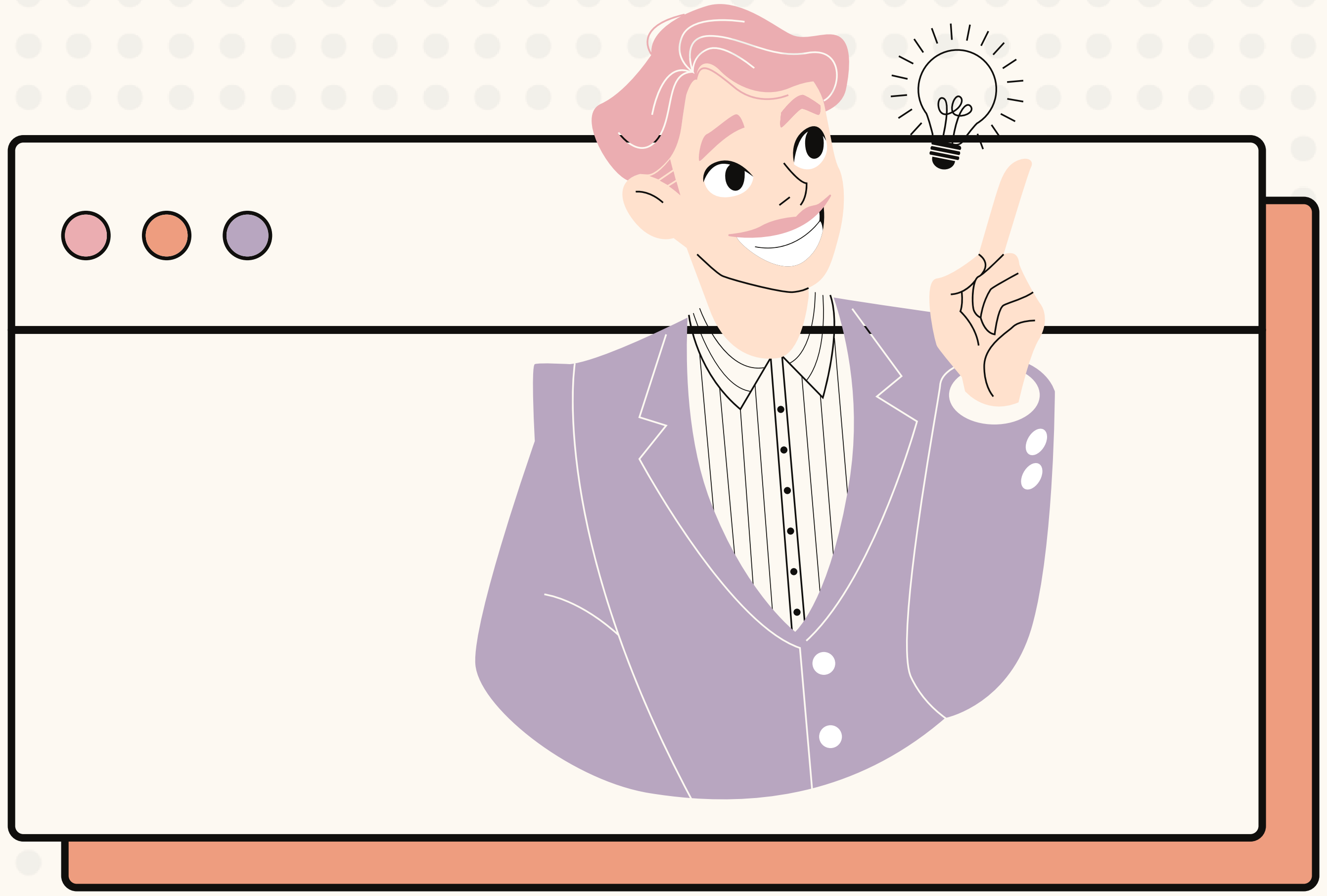
Participantes

ALINE

ANANDA

JULIANA

KHÉZIA



Introdução

Diagram: utilizado para a elaboração de gráficos simples baseados em matrizes de transição, os "grafos".

Markovchain: disponibiliza classes, métodos e funções das Cadeias de Markov Discretas.

Passos

Criar a matriz → Criar a C.M. utilizando o markovchain → Plotar o grafo da C.M. utilizando o diagram → Análises estatísticas da C.M por meio do markchain

Entendendo o markvochain

O pacote traz várias funções que permitem análisar e realizar cálculos.

O argumento principal das funções é **o objeto da classe markovchain**, que precisa ser definido primeiramente.

Definindo o argumento (o objeto da classe)

1 CRIANDO A MATRIZ DE TRANSIÇÃO

```
matriz_transicao <- matrix(c(0.8, 0, 0, 0, 0, 0.2, 0, 0, 0, 0,
0, 1, 0, 0, 0.1, 0, 0.9, 0,
0, 0, 0, 0, 0, 0, 0.5, 0, 0, 0.5, 0, 0.3, 0, 0, 0.7, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0, 0.5, 0,
0, 0, 0.5, 0), nrow=7, byrow=TRUE,
dimnames=list(c('0', '1', '2', '3', '4', '5', '6'), c('0', '1',
'2', '3', '4', '5', '6')))
```

1 CRIANDO A CADEIA DE MARKOV

```
cadeia_markov <- new("markovchain", states=c('0', '1', '2', '3',
'4', '5', '6'),
transitionMatrix=matriz_transicao)

cadeia_markov
```

Definindo o argumento (o objeto da classe)

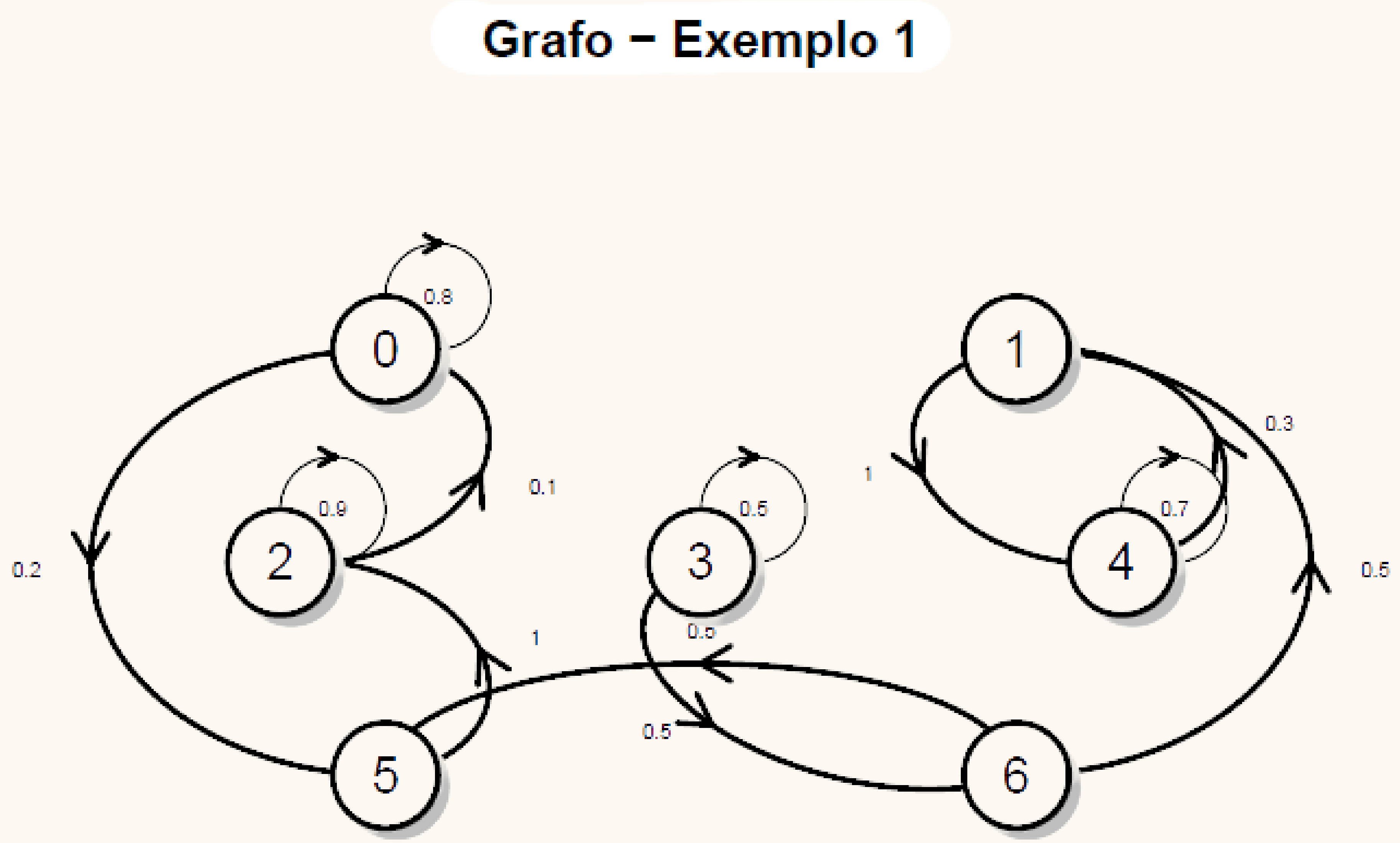
```
cadeia_markov
```

```
## Unnamed Markov chain
##  A  7 - dimensional discrete Markov Chain defined by the following
##  0, 1, 2, 3, 4, 5, 6
##  The transition matrix  (by rows)  is defined as follows:
##     0   1   2   3   4   5   6
## 0 0.8 0.0 0.0 0.0 0.0 0.2 0.0
## 1 0.0 0.0 0.0 0.0 1.0 0.0 0.0
## 2 0.1 0.0 0.9 0.0 0.0 0.0 0.0
## 3 0.0 0.0 0.0 0.5 0.0 0.0 0.5
## 4 0.0 0.3 0.0 0.0 0.7 0.0 0.0
## 5 0.0 0.0 1.0 0.0 0.0 0.0 0.0
## 6 0.0 0.5 0.0 0.0 0.0 0.5 0.0
```

Plotando o grafo

3 **CRIANDO O GRAFO:** FUNÇÃO PLOTMAT() DO DIAGRAM

```
plotmat(t(matriz_transicao), pos=c(2, 3, 2), curve = 0.35,
arr.len = 0.3,arr.type = "simple", dtext = 0.95, box.size =
0.05, cex.txt = 0.45, relsize=0.8, main="Grafo - Exemplo 1")
```



Plotando o grafo

3 **CRIANDO O GRAFO:** FUNÇÃO PLOTMAT() DO DIAGRAM

```
plotmat(t(matriz_transicao), pos=c(2, 3, 2), curve = 0.35,
arr.len = 0.3, arr.type = "simple", dtext = 0.95, box.size =
0.05, cex.txt = 0.45, relsize=0.8, main="Grafo - Exemplo 1")
```

controla o tamanho das setas que conectam os grafos

Vetor indicando quantos estados teremos por linha de forma ordenada

controla a angulação das setas que conectam os grafos;

determina o formato da seta, no nosso caso, é simples

controla a posição do texto da seta em relação à ponta da seta

Plotando o grafo

3 CRIANDO O GRAFO: FUNÇÃO PLOTMAT() DO DIAGRAM

```
plotmat(t(matriz_transicao), pos=c(2, 3, 2), curve = 0.35,
arr.len = 0.3,arr.type = "simple", dtext = 0.95, box.size =
0.05, cex.txt = 0.45, relsize=0.8, main="Grafo - Exemplo 1")
```

controla o tamanho do texto que acompanha a seta

controla a escala do gráfico

define o título do gráfico.

controla o tamanho da caixa dos estados;

Realizando operações

4 REALIZANDO OPERAÇÕES

VENDO OS ESTADOS: FUNÇÃO NAMES()

```
names(cadeia_markov)
```

```
## [1] "0" "1" "2" "3" "4" "5" "6"
```

VENDO OS ESTADOS: FUNÇÃO STATES()

```
states(cadeia_markov)
```

```
## [1] "0" "1" "2" "3" "4" "5" "6"
```

Realizando operações

CALCULANDO A MATRIZ DE N PASSOS:

```
P_3 <-cadeia_markov^3
P_3
```

```
## Unnamed Markov chain^3
##  A  7 - dimensional discrete Markov Chain defined by the following states:
##  0, 1, 2, 3, 4, 5, 6
##  The transition matrix  (by rows)  is defined as follows:
##        0     1     2     3     4     5     6
## 0 0.532 0.000 0.340 0.000 0.000 0.128 0.000
## 1 0.000 0.210 0.000 0.000 0.790 0.000 0.000
## 2 0.217 0.000 0.749 0.000 0.000 0.034 0.000
## 3 0.000 0.125 0.250 0.125 0.250 0.125 0.125
## 4 0.000 0.237 0.000 0.000 0.763 0.000 0.000
## 5 0.170 0.000 0.810 0.000 0.000 0.020 0.000
## 6 0.050 0.150 0.450 0.000 0.350 0.000 0.000
```

Realizando operações

VERIFICANDO SE A CADEIA POSSUI ESTADOS ABSORVENTES: ABSORBINGSTATES ( )

```
absorbingStates(cadeia_markov)
```

```
## character(0)
```

VERIFICANDO SE A CADEIA POSSUI ESTADOS RECORRENTES: RECURRENTSTATES( )

```
recurrentStates(cadeia_markov)
```

```
## [1] "0" "1" "2" "4" "5"
```

VERIFICANDO SE A CADEIA POSSUI ESTADOS TRANSIENTES: TRANSIENTSTATES( )

```
transientStates(cadeia_markov)
```

```
## [1] "3" "6"
```

Realizando operações

VERIFICANDO AS CLASSES DE CADEIA DA CADEIA: COMMUNICATINGCLASSES( )

```
communicatingClasses(cadeia_markov)
```

```
## [[1]]
## [1] "0" "2" "5"
##
## [[2]]
## [1] "1" "4"
##
## [[3]]
## [1] "3"
##
## [[4]]
## [1] "6"
```

TEMOS ENTÃO AS CLASSES {0, 2, 5}, {1, 4}, {3} E {6}

Realizando operações

CLASSIFICANDO AS CLASSES RECORRENTES: RECURRENTCLASSES()

```
recurrentClasses(cadeia_markov)
```

```
## [[1]]
## [1] "0" "2" "5"
##
## [[2]]
## [1] "1" "4"
```

CLASSIFICANDO AS CLASSES TRANSIENTES: TRANSIENTCLASSES()

```
transientClasses(cadeia_markov)
```

```
## [[1]]
## [1] "3"
##
## [[2]]
## [1] "6"
```

Realizando operações

AGRUPANDO AS CLASSES POR TIPO

```
canonicForm(cadeia_markov)
```

```
## Unnamed Markov chain
##  A  7 - dimensional discrete Markov Chain defined by the following states:
##  0, 2, 5, 1, 4, 3, 6
##  The transition matrix  (by rows)  is defined as follows:
##      0   2   5   1   4   3   6
## 0 0.8 0.0 0.2 0.0 0.0 0.0 0.0
## 2 0.1 0.9 0.0 0.0 0.0 0.0 0.0
## 5 0.0 1.0 0.0 0.0 0.0 0.0 0.0
## 1 0.0 0.0 0.0 0.0 1.0 0.0 0.0
## 4 0.0 0.0 0.0 0.3 0.7 0.0 0.0
## 3 0.0 0.0 0.0 0.0 0.0 0.5 0.5
## 6 0.0 0.0 0.5 0.5 0.0 0.0 0.0
```

MOSTRA PRIMEIRO AS CLASSES RECORRENTES E DEPOIS AS CLASSES TRANSIENTES.

Realizando operações

CALCULANDO O f_{ij}^{n}

```
firstPassage(cadeia_markov, '0', 10)
```

```
##              0 1          2 3 4          5 6
## 1  0.800000000 0 0.00000000 0 0 0.20000000 0
## 2  0.000000000 0 0.20000000 0 0 0.16000000 0
## 3  0.020000000 0 0.16000000 0 0 0.12800000 0
## 4  0.018000000 0 0.12800000 0 0 0.10240000 0
## 5  0.016200000 0 0.10240000 0 0 0.08192000 0
## 6  0.014580000 0 0.08192000 0 0 0.06553600 0
## 7  0.013122000 0 0.06553600 0 0 0.05242880 0
## 8  0.011809800 0 0.05242880 0 0 0.04194304 0
## 9  0.010628820 0 0.04194304 0 0 0.03355443 0
## 10 0.009565938 0 0.03355443 0 0 0.02684355 0
```

i=0 e
n=1, 2, 3, 4, 5,
6, 7, 8, 9, 10.

Realizando operações

CALCULANDO AS PROBABILIDADES DAS LINHAS

```
conditionalDistribution(cadeia_markov, '0')
```

```
##   0   1   2   3   4   5   6
## 0.8 0.0 0.0 0.0 0.0 0.2 0.0
```

CHECANDO SE 1 ACESSA 2

```
is.accessible(cadeia_markov, '1', '2') #i=1 e j=2
```

```
## [1] FALSE
```

CHECANDO SE 2 ACESSA 1

```
is.accessible(cadeia_markov, '2', '1') #i=2 e j=1
```

```
## [1] FALSE
```

Realizando operações

CHECANDO IRREDUTIBILIDADE

```
is.irreducible(cadeia_markov)
```

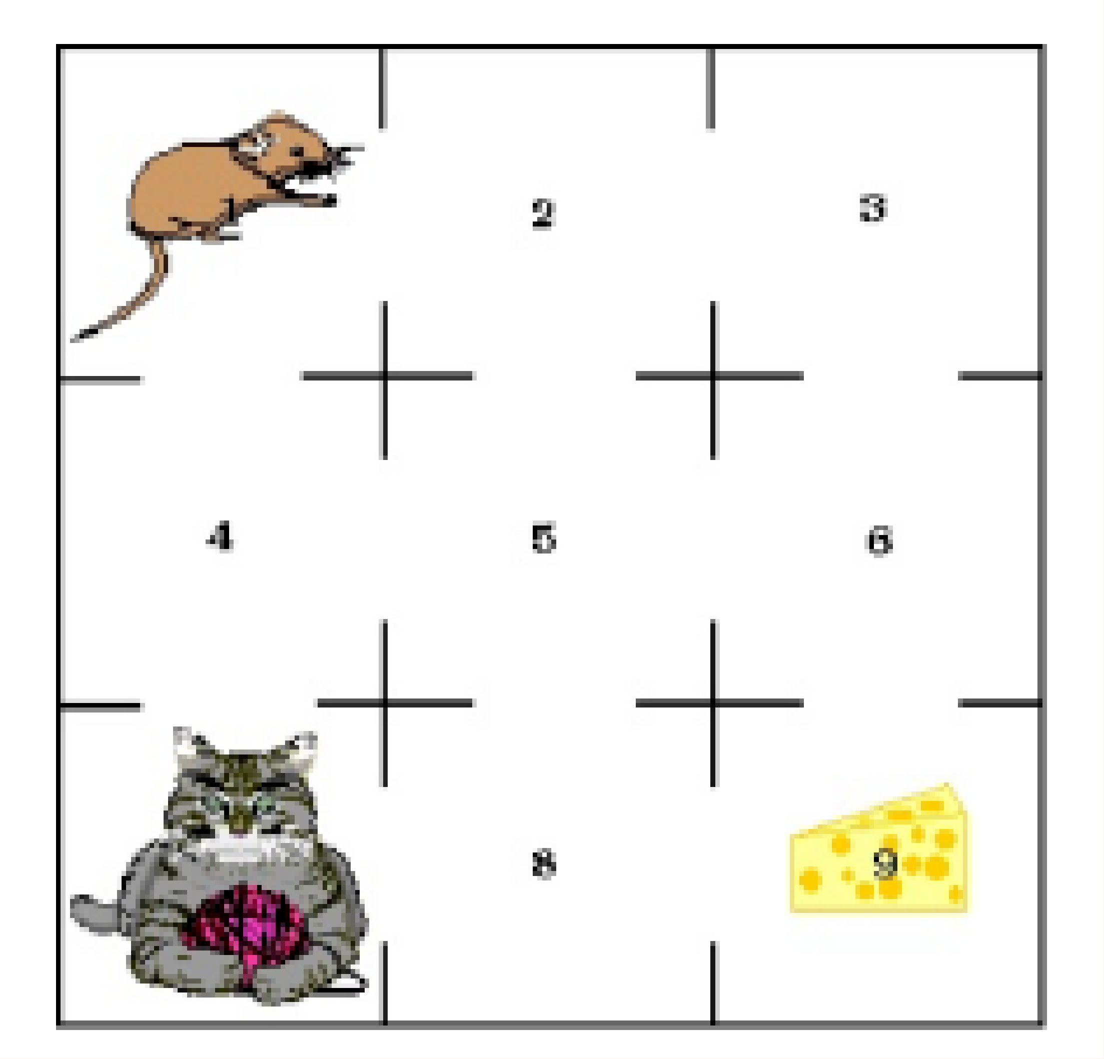
```
## [1] FALSE
```

OBTENDO O NÚMERO MÉDIO DE VISITAS AO ESTADO J PARTINDO DO ESTADO I

```
meanNumVisits(cadeia_markov)
```

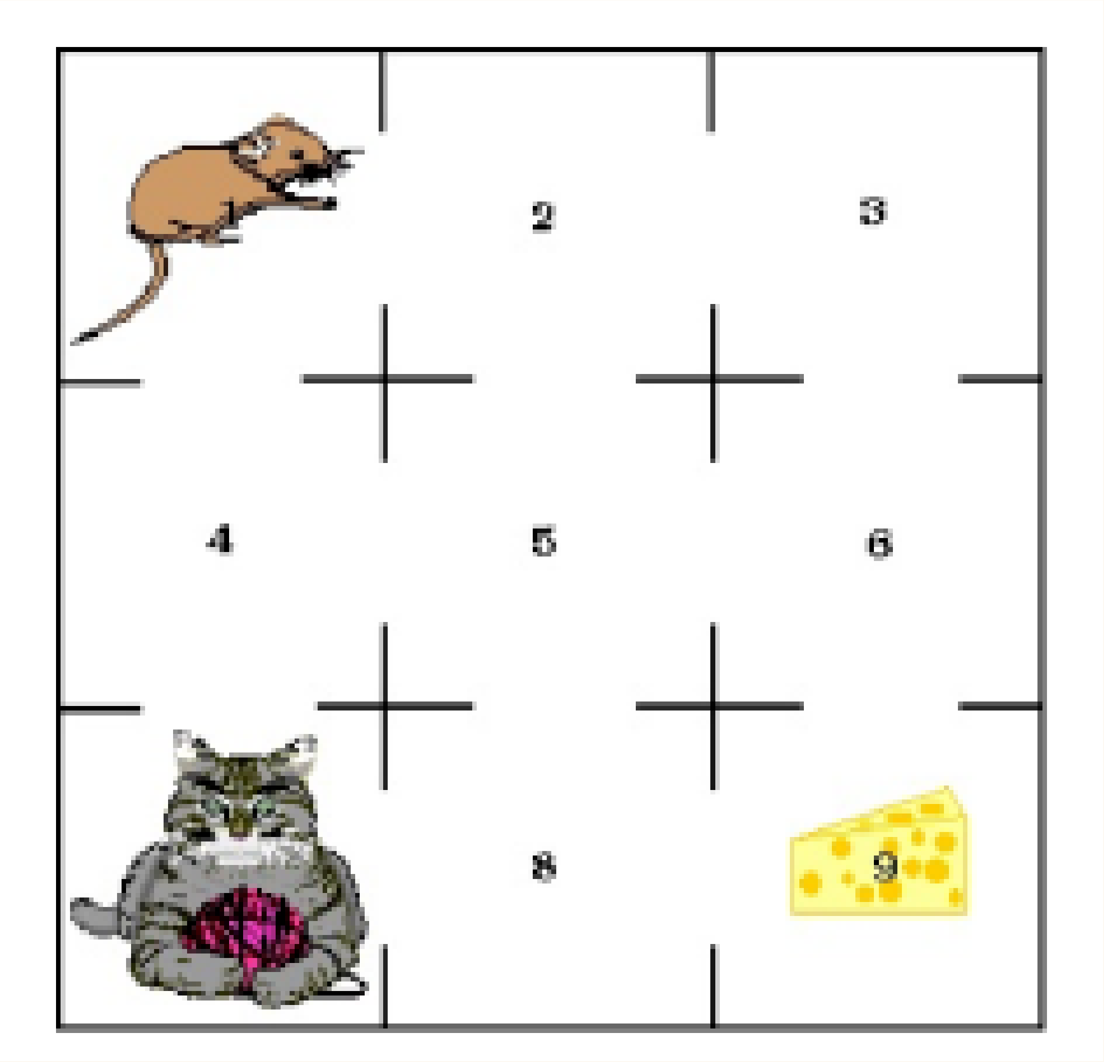
```
##       0   1   2 3   4   5 6
## 0 Inf   0 Inf 0   0 Inf 0
## 1   0 Inf   0 0 Inf   0 0
## 2 Inf   0 Inf 0   0 Inf 0
## 3 Inf Inf Inf 1 Inf Inf 1
## 4   0 Inf   0 0 Inf   0 0
## 5 Inf   0 Inf 0   0 Inf 0
## 6 Inf Inf Inf 0 Inf Inf 0
```

Exemplo do Rato no labirinto



Considere que um rato é colocado em um dos compartimentos de um labirinto como apresentado na figura que está abaixo. O rato move-se aleatoriamente entre os compartimentos com mesma probabilidade e o estado do sistema no instante n é o número do compartimento em que ele se encontra neste instante.

Exemplo do Rato no labirinto



$$P = \begin{bmatrix} 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 0 & 1/3 & 0 & 0 \\ 0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 1/3 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 1/3 & 0 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 \end{bmatrix}$$

Rato no labirinto

CALCULANDO A MATRIZ DE N PASSOS:

```
matriz_rato <- matrix(c(0, 0.5, 0, 0.5, 0, 0, 0, 0, 0, 1/3, 0,
1/3, 0, 1/3, 0, 0, 0, 0, 0, 0.5, 0, 0, 0, 0.5, 0, 0, 0, 1/3, 0,
0, 0, 1/3, 0, 1/3, 0, 0, 0, 0.25, 0, 0.25, 0, 0.25, 0, 0.25, 0,
0, 0, 1/3, 0, 1/3, 0, 0, 0, 1/3, 0, 0, 0, 0.5, 0, 0, 0, 0.5, 0,
0, 0, 0, 0, 1/3, 0, 1/3, 0, 1/3, 0, 0, 0, 0, 0, 0.5, 0, 0.5, 0),
nrow=9, byrow=TRUE,
dimnames=list(c('1', '2', '3', '4', '5', '6', '7', '8', '9'),
c('1', '2', '3', '4', '5', '6', '7', '8', '9')))

cadeia_markov_rato <- new("markovchain", states=c('1', '2', '3',
'4', '5', '6', '7', '8', '9'), transitionMatrix=matriz_rato)
```

Rato no labirinto

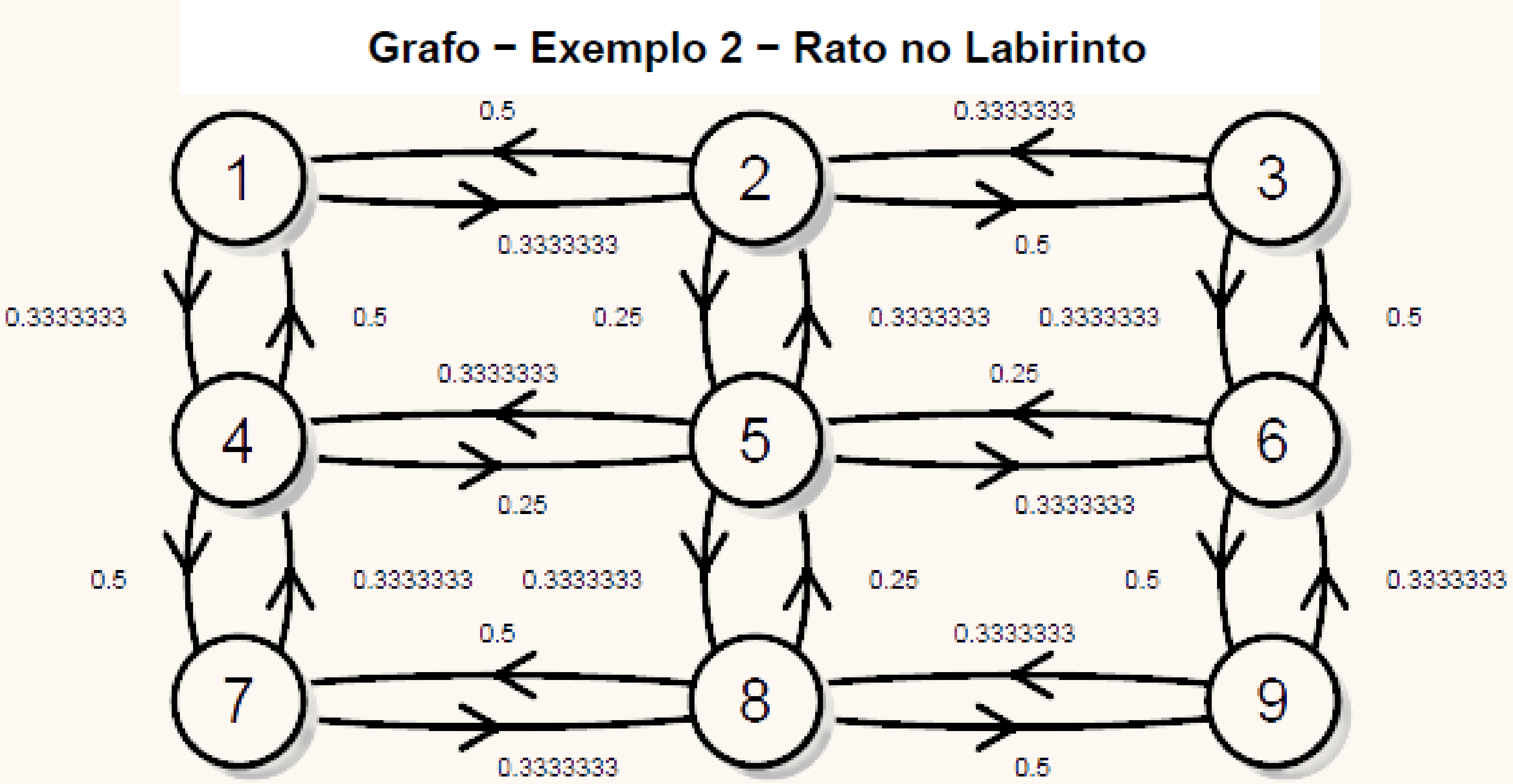
```
cadeia_markov_rato
```

```
## Unnamed Markov chain
##  A  9 - dimensional discrete Markov Chain defined by the following states:
##  1, 2, 3, 4, 5, 6, 7, 8, 9
##  The transition matrix  (by rows)  is defined as follows:
##           1    2         3    4         5    6         7    8         9
## 1 0.0000000 0.50 0.0000000 0.50 0.0000000 0.00 0.0000000 0.00 0.0000000
## 2 0.3333333 0.00 0.3333333 0.00 0.3333333 0.00 0.0000000 0.00 0.0000000
## 3 0.0000000 0.50 0.0000000 0.00 0.0000000 0.50 0.0000000 0.00 0.0000000

## 4 0.3333333 0.00 0.0000000 0.00 0.3333333 0.00 0.3333333 0.00 0.0000000
## 5 0.0000000 0.25 0.0000000 0.25 0.0000000 0.25 0.0000000 0.25 0.0000000
## 6 0.0000000 0.00 0.3333333 0.00 0.3333333 0.00 0.0000000 0.00 0.3333333
## 7 0.0000000 0.00 0.0000000 0.50 0.0000000 0.00 0.0000000 0.50 0.0000000
## 8 0.0000000 0.00 0.0000000 0.00 0.3333333 0.00 0.3333333 0.00 0.3333333
## 9 0.0000000 0.00 0.0000000 0.00 0.0000000 0.50 0.0000000 0.50 0.0000000
```

Rato no labirinto

CRINDO O GRAFO

```
AA <- as.data.frame(matriz_rato)

plotmat(A = AA, pos = c(3, 3, 3), curve = 0.1, arr.len = 0.3, arr.type
= "simple", dtext = 0.95, box.size = 0.05, cex.txt = 0.45, relsize=0.8,
main = "Grafo - Exemplo 2 - Rato no Labirinto")
```



Realizando operações

CONFERIR SE ESSA É UMA CADEIA IRREDUTÍVEL:

```
is.irreducible(cadeia_markov_rato)
```

```
## [1] TRUE
```

VENDO AS CLASSES RECORRENTES

```
recurrentClasses(cadeia_markov_rato)
```

```
## [[1]]
## [1] "1" "2" "3" "4" "5" "6" "7" "8" "9"
```

VENDO O PERÍODO DA CADEIA

```
period(cadeia_markov_rato)
```

```
## [1] 2
```

Rato no labirinto

DESCOBRIRMOS A MATRIZ DE TRANSIÇÃO A NOVE PASSOS

```
cadeia_9_passos <- cadeia_markov_rato ^ 9
cadeia_9_passos
```

```
## Unnamed Markov chain^9
##  A  9 - dimensional discrete Markov Chain defined by the following states:
##  1, 2, 3, 4, 5, 6, 7, 8, 9
##  The transition matrix  (by rows)  is defined as follows:
##           1         2         3         4         5         6         7
## 1 0.0000000 0.2530864 0.0000000 0.2530864 0.0000000 0.2469136 0.0000000
## 2 0.1687243 0.0000000 0.1687243 0.0000000 0.3333333 0.0000000 0.1646091
## 3 0.0000000 0.2530864 0.0000000 0.2469136 0.0000000 0.2530864 0.0000000
## 4 0.1687243 0.0000000 0.1646091 0.0000000 0.3333333 0.0000000 0.1687243
## 5 0.0000000 0.2500000 0.0000000 0.2500000 0.0000000 0.2500000 0.0000000
## 6 0.1646091 0.0000000 0.1687243 0.0000000 0.3333333 0.0000000 0.1646091
## 7 0.0000000 0.2469136 0.0000000 0.2530864 0.0000000 0.2469136 0.0000000
## 8 0.1646091 0.0000000 0.1646091 0.0000000 0.3333333 0.0000000 0.1687243
## 9 0.0000000 0.2469136 0.0000000 0.2469136 0.0000000 0.2530864 0.0000000
##           8         9
## 1 0.2469136 0.0000000
## 2 0.0000000 0.1646091
## 3 0.2469136 0.0000000
## 4 0.0000000 0.1646091
## 5 0.2500000 0.0000000
## 6 0.0000000 0.1687243
## 7 0.2530864 0.0000000
## 8 0.0000000 0.1687243
## 9 0.2530864 0.0000000
```

Rato no labirinto

COM A DIST. INICIAL

```
# Definindo alpha0
a_0 <- c(1/9,1/9,1/9,1/9,1/9,1/9,1/9,1/9,1/9)

# Definindo a coluna 6 de p^9
coluna6 <- cadeia_9_passos[,6]

# Fazendo o cálculo
sum(a_0 * coluna6)
```

```
## [1] 0.1388889
```

A PROBABILIDADE DE QUE O RATO ESTEJA NO COMPARTIMENTO 6 DEPOIS DE 9 TRANSIÇÕES É DEAPROXIMADAMENTE 13,89%.

Mosca vs. Aranha

Uma aranha, tentando caçar uma mosca, move-se entre as localizações 1 e 2 de acordo com uma Cadeia de Markov com matriz de transição dada a seguir:

$$Q = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}$$

Por outro lado, a mosca que não se dá conta da presença da aranha, move-se de acordo com uma Cadeia de Markov com a seguinte matriz de transição:

$$R = \begin{bmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{bmatrix}$$

Mosca vs. Aranha

PODEMOS CONSIDERAR 3 ESTADOS:

- A1M2 PARA QUANDO A ARANHA ESTÁ EM 1 E A MOSCA EM 2;
- A2M1 PARA QUANDO A ARANHA ESTÁ EM 2 E A MOSCA EM 1;
- FIM PARA QUANDO A CAÇADA TERMINA (ARANHA E MOSCA ESTÃO NA MESMA POSIÇÃO - AMBAS EM 1 OU AMBAS EM 2).

REALIZANDO CÁLCULOS

- SE A ARANHA COMEÇA EM 1 (A1) E A MOSCA EM 2 (M2):

PROBABILIDADE DAS DUAS CONTINUAREM NESSAS POSIÇÕES

```
P_A1M2_A1M2 <- 0.7*0.4  #aranha continua em 1 * mosca continua em 2
```

PROBABILIDADE DAS DUAS TROCAREM DE POSIÇÕES

```
P_A1M2_A2M1 <- 0.3*0.6 #aranha muda para 2 * mosca muda para 1
```

PROBABILIDADE DE QUE A ARANHA PEGUE A MOSCA

```
P_A1M2_Fim <- 0.7*0.6+0.3*0.4 #aranha continua em 1 * mosca muda para 1
#+ aranha muda para 2 * mosca continua em 2
```

Mosca vs. Aranha

OBTEMOS ASSIM OS RESULTADOS DA LINHA 1 DA MATRIZ DE TRANSIÇÃO

```
linha1 <- c(P_A1M2_A1M2, P_A1M2_A2M1, P_A1M2_Fim)
```

```
## [1] 0.28 0.18 0.54
```

REALIZANDO CÁLCULOS

- SE A ARANHA COMEÇA EM 2(A2) E A MOSCA EM 1 (M1)

PROBABILIDADE DE AS DUAS CONTINUAREM NESSAS POSIÇÕES

```
P_A2M1_A2M1 <- 0.7*0.4 #aranha continua em 2 * mosca continua em 1
```

PROBABILIDADE DE AS DUAS TROCAREM DE POSIÇÕES

```
P_A2M1_A1M2 <- 0.3*0.6 #aranha muda para 1 * mosca muda para 2
```

PROBABILIDADE DE QUE A ARANHA PEGUE A MOSCA

```
P_A2M1_Fim <- 0.7*0.6+0.3*0.4 #aranha continua em 2 * mosca muda para 2
#+ aranha muda para 1 * mosca continua em 1
```

Mosca vs. Aranha

OBTEMOS ASSIM OS RESULTADOS DA A LINHA 2 DA MATRIZ DE TRANSIÇÃO

```
linha2 <- c(P_A2M1_A1M2, P_A2M1_A2M1, P_A2M1_Fim)
linha2
```

```
## [1] 0.18 0.28 0.54
```

E POR FIM, A TERCEIRA LINHA É QUANDO PARTIMOS DO NOSSO TERCEIRO ESTADO, QUE É "FIM". NESSE CASO, A CAÇADA JÁ ACABOU, ENTÃO AS PROBABILIDADES DE MUDAR PARA OS OUTROS ESTADOS SÃO NULAS E A PROBABILIDADE DE PERMANECER NO ESTADO "FIM" É UNITÁRIA.

```
linha3 <- c(0, 0, 1)
linha3
```

Mosca vs. Aranha

DAÍ, TEMOS FINALMENTE A NOSSA CADEIA:

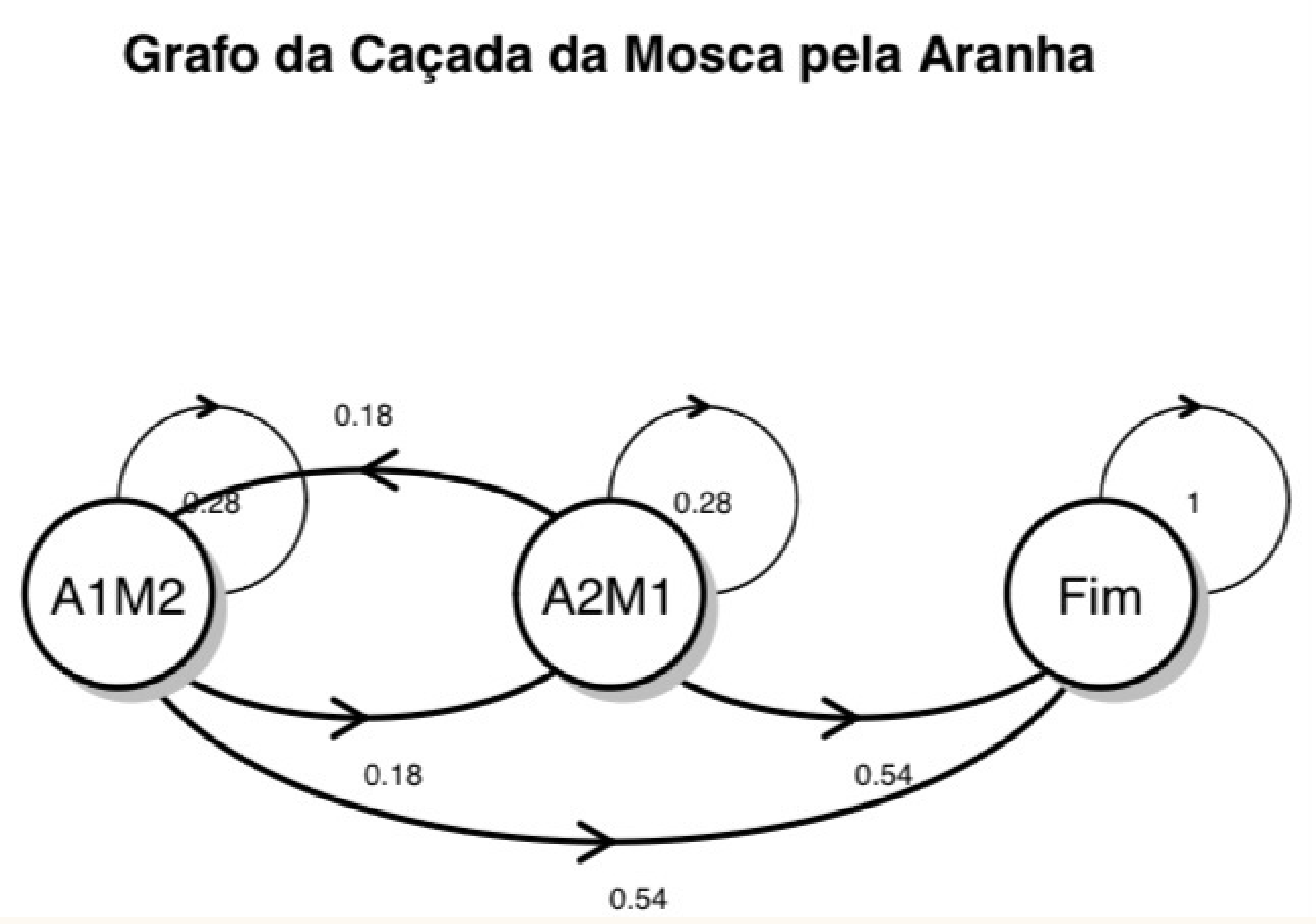
```
matriz_cacada <- matrix(append(append(linha1, linha2), linha3),
nrow=3, byrow=TRUE,
dimnames=list(c("A1M2", "A2M1", "Fim"), c("A1M2", "A2M1",
"Fim")))

#Criando a cadeia de markov
cadeia_markov_cacada <- new("markovchain", states=c("A1M2",
"A2M1", "Fim"),transitionMatrix=matriz_cacada)

cadeia_markov_cacada
```

```
## Unnamed Markov chain
##  A  3 - dimensional discrete Markov Chain defined by the following states:
##  A1M2, A2M1, Fim
##  The transition matrix  (by rows)  is defined as follows:
##      A1M2 A2M1  Fim
## A1M2 0.28 0.18 0.54
## A2M1 0.18 0.28 0.54
## Fim  0.00 0.00 1.00
```

Mosca vs. Aranha

FAZENDO O GRAFO:

```
plotmat(t(matriz_cacada), pos = 3, curve = 0.5, arr.len = 0.3,
arr.type = "simple", dtext = 0.95, box.size = 0.07, cex.txt =
0.55,relsize=0.9, main="Grafo da Caçada da Aranha pela Mosca")
```



Obrigada!