

- [Tutorial](#)
- [Exercícios](#)
- [Apostila](#)

3. Tutoriais de Leitura e Manipulação de dados no R

Vetores como estrutura básica de dados

Nos tutoriais anteriores, nós usamos o objeto que é a estrutura básica dos objetos de dados, a classe de objetos vetores. Crie um vetor de dados numéricos, chamado "num", que seja uma sequência de valores de 0 a 15 divididos em 7 valores equidistantes.

```
num <- seq(0,15, len=7)
```

Agora vamos olhar os valores do objeto criado, em seguida elevar cada valor ao quadrado, atribuindo o resultado ao objeto "num2". Em seguida tire a raiz-quadrada desse vetor.

```
num
length(num)
class(num)
num2<- num^2
num2
sqrt(num2)
num
```

Agora vamos extrair apenas o quinto elemento e em seguida substituir esse elemento pela palavra "quinto_elemento". Em seguida vamos repetir as operações anteriores:

```
num[5]
num[5]<- "quinto_elemento"
num
num[-5]

length(num)
class(num)
num^2
sqrt(num^2)
```

O que aconteceu aqui? Por que não conseguimos mais operar o vetor "num"? Vamos agora voltar o vetor para sua forma original e operar novamente:

```
num[5]<- 10
num
num^2
```

```
sqrt(num^2)
```

Por que ainda não conseguimos? Vamos investigar o vetor e ajustá-lo.

```
length(num)
class(num)
num <- as.numeric(num)
sqrt(num^2)
```

A raiz de um valor elevado ao quadrado é o próprio valor? O que aconteceria se o vetor fosse de números negativos? Veja a discussão sobre função quadrática e raiz quadrada como [funções inversas nesse link](#)

Leitura de Dados

Salve o arquivo de dados de [levantamento de espécies em caixetais](#) no seu diretório de trabalho e crie um objeto com seu conteúdo em sua área de trabalho do R. Usaremos este conjunto de dados em todos os tutoriais da unidade, exceto o último.

Obs.: Para salvar, clique com o botão direito do mouse e selecione “salvar linque” 😊.

Depois de efetuar a leitura **sempre verifique a estrutura dos dados antes de continuar suas análises.**

```
caixeta = read.table("caixeta.csv", header=T, sep="," , as.is=T)
class(caixeta)
dim(caixeta)
names(caixeta)
str(caixeta)
head(caixeta)
```

Transforme a coluna especie em fator, em um vetor separado. Isso ajuda na manipulação de dados, já que o vetor é correspondente à coluna, mas é de uma classe diferente.

```
spp=factor(caixeta$especie)
spp
```

Verifique o resultado dos comandos abaixo

```
unique(spp)
unique(caixeta$especie)
```

Tabelas e Matrizes

Prosseguindo com o mesmo conjunto de dados do tutorial anterior:

Contando spp por local e selecionando uma localidade

```
str(caixeta)
caixeta.local = table(caixeta$especie, caixeta$local)
str(caixeta.local)
class(caixeta.local)
dimnames(caixeta.local)
caixeta.local[, "chauas"]
```

Transformando um objeto de outra classe em matriz

```
caixeta.mat=matrix(caixeta.local,ncol=3)
colnames(caixeta.mat)<-colnames(caixeta.local)
rownames(caixeta.mat)<-rownames(caixeta.local)
caixeta.mat
str(caixeta.mat)
```

Faz diferença??

```
table(caixeta.local==caixeta.mat)
```

Totais Marginais

Número de espécies por local

```
caixeta.vf=caixeta.mat>0
caixeta.vf
riqueza=apply(caixeta.vf,2,sum)
riqueza
```

Indexação e Manipulação

```
chauas = caixeta.mat[, "chauas"]
which(chauas>0)
chauas.bin <- chauas
chauas.bin[which(chauas>0)]<-1
chauas.bin
```

Selecionando colunas, linhas e valores do objeto caixeta, baseado em critérios de algum estado de um fator (espécie, local).

```
str(caixeta)
nome.spp <- unique(caixeta$especie)
nome.local <- unique(caixeta$local)
caixeta[caixeta$especie==nome.spp[1],]
caixeta[caixeta$local==nome.local[2],]
caixeta[caixeta$especie==nome.spp[1] & caixeta$local==nome.local[2],]
```

Qual o erro?

Tente esse comando:

```
caixeta[caixeta$especie==nome.spp[1] & caixeta$local==nome.local[1],]
```

Usando matrizes para simular

Se tomamos uma amostra ao acaso com reposição¹⁾ de uma população estatística e medimos algum atributo dos elementos desta amostra, podemos estimar a variância desta medida na população com :

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}$$

Onde

- s^2 é o estimador da variância
- x_i é cada uma das medidas, tomadas do elemento $i=1$ ao n -ésimo
- $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ é a média das medidas da amostra
- n é o número de elementos na amostra (tamanho amostral)

O denominador $n-1$ é necessário para que o estimador não seja enviesado. O que aconteceria se usássemos o tamanho da amostra n e não $n-1$ como denominador?

Vamos investigar isto com uma simulação, sorteando amostras de uma população com uma variância conhecida, e calculamos a variância de cada uma delas pelos dois métodos. Um estimador sem viés em média tem o valor do que ele estima, que é parâmetro da população que ele estima. Portanto, basta comparar as médias das variâncias calculadas das duas maneiras para avaliar se há viés.

Usaremos as funções de construção de matrizes e de cálculos de funções marginais (`apply`) para realizar estas simulações.

Para calcular a variância amostral definida pela expressão acima temos a função `var` do R. Vamos agora criar uma função para calcular a variância sem a correção de viés $n-1$, ou seja:

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$$

Para isto copie e cole o seguinte comando no R ²⁾:

```
var.2=function(x) (var(x)*(length(x)-1))/length(x)
```

Agora vamos criar mil amostras de tamanho 10, tomadas de uma população com distribuição normal de média igual a um e variância igual a quatro. O primeiro passo é gerar dez mil valores sorteados desta distribuição normal:

```
valores = rnorm(10000, mean=1, sd=2)
```

Agora vamos distribuir estes valores em uma matriz com 1000 colunas. Como são 10.000 valores, cada coluna terá 10 linhas, e representará uma das 1000 amostras de tamanho 10.

```
valores.m = matrix(valores, ncol=1000)
```

Agora vamos usar a função `apply` para calcular as variâncias de cada coluna desta matriz, usando as duas funções

```
##Estimador com correção
variancias1 = apply(valores.m, 2, var)
## Estimador sem correção
variancias2 = apply(valores.m, 2, var.2)
```

Por fim calculamos as médias dos dois conjuntos de 1000 variâncias. O estimador sem viés terá uma média que converge ao valor da variância na população, que é 4:

```
mean(variancias1)
mean(variancias2)
```

Operações Matriciais

✘ As matrizes de transição são uma maneira conveniente de modelar o crescimento de uma população dividida em faixas etárias, ou estágios de desenvolvimento ³⁾. Uma forma de representar essas populações que apresentam estágios diferentes é com diagramas mostrando as transições e permanências entre estados.

Uma forma compacta de representar matematicamente essas transições e permanência é utilizando a forma de matriz, que nesse caso se chama **matriz de transição**, como na figura abaixo: ✘

Para uma população de *Coryphanta robinsorum* (Cactaceae) no deserto do Arizona, dividida em três estágios, a matriz de transição foi:

0,43	0	0,56
0,33	0,61	0
0	0,30	0,96

Os elementos da matriz são a probabilidades de transição, num intervalo de tempo, do estágio correspondente ao número da coluna para o estágio correspondente ao número da linha. Por exemplo, a chance de um indivíduo passar do estágio 1 (plântula) para o 2 (jovem) é 0,33, e de permanecer em 1 é de 0,43.

Crie um objeto da classe `matriz` com esses valores. Isso nos permitirá realizar operações matriciais para prever o estado da população.

```
mat.trans <- matrix(c(0.43,0.33,0,0,0.61,0.3,0.56,0,0.96),3,3)
mat.trans
```

Agora crie um vetor com as abundâncias iniciais de indivíduos em cada classe. Vamos começar com 50 plântulas, 25 jovens e 10 adultos.

```
pop.inicio <- c(50,25,10)
names(pop.inicio)<-c("plântula", "jovem", "adulto")
pop.inicio
```

Para calcular o número de indivíduos em cada estágio após um intervalo de tempo, basta multiplicar a matriz de transição pelas abundâncias dos indivíduos em cada estágio. Usamos o operador de multiplicação matricial `%*%` para isso. Qual será o número de plantas em cada estágio após três intervalos?

```
pop.1 <- mat.trans%*%pop.inicio
pop.1
pop.2 <- mat.trans%*%pop.1
pop.2
pop.3 <- mat.trans%*%pop.2
pop.3
```

Vamos agora armazenar a trajetória do tamanho dos estágios na população em um data frame:

```
pop <- data.frame(t0=pop.inicio, t1=pop.1, t2=pop.2, t3=pop.3)
pop
```

Por fim, vamos olhar esses dados em um gráfico. O código abaixo vai ser abordado na aula de gráficos, por enquanto apenas copie e cole as linhas no console do R:

```
matplot(x=0:3, t(pop), type="l", xlab="tempo", ylab="numero de indivíduos")
legend("topright", legend=names(pop.inicio), bty="n", lty=1:3, col=1:3)
```

Opcional: a multiplicação de matrizes usados neste tipo de modelo de dinâmica populacional estruturada. Com um pouco mais álgebra linear você pode obter muito mais informações. Por exemplo, a taxa de crescimento geométrico da população é o primeiro autovalor da matriz de transição, que pode ser calculado com a função `eigen`⁴⁾. Se a taxa é maior que um a população está crescendo. É o caso dessa população?

```
eigen(mat.trans)
```

¹⁾ ou uma amostra muito pequena em relação ao tamanho da população, pois se aproxima de uma amostra com reposição

²⁾ Você irá entender este comando na [unidade sobre criação de funções](#). Por ora basta aceitar que criamos uma função que calcula a variância sem a correção de viés. Como trata-se apenas de aceitar, pode copiar e colar o comando. Mas em geral digite os comandos dos tutoriais

³⁾ para saber mais veja a [Página da disciplina de Ecologia Vegetal](#)

⁴⁾ consulte a ajuda para interpretar o resultados dessa função

From:

<http://labtrop.ib.usp.br/> - **Laboratório de Ecologia de Florestas Tropicais**

Permanent link:

http://labtrop.ib.usp.br/doku.php?id=cursos:ecor:02_tutoriais:tutorial3:start



Last update: **2020/07/27 18:49**