

INTRODUÇÃO AOS MODELOS LINEARES EM ECOLOGIA

PROF. ADRIANO SANCHES MELO – asm.adrimelo no gmail.com -DEP. ECOLOGIA – UFG

PÁGINA DO CURSO: www.ecoevol.ufg.br/adrimelo/lm/

AULA 2– COMANDOS NO R

O PACOTE R

Diretório de trabalho

Menu

Pacotes

Linhas de comando: setas para voltar

Tipos de objetos: `vector`, `matrix`, `array`, `factor`, `data.frame`, `list`, `function`

0. Informações gerais:

--Use ponto "." para casas decimais.

--Exceto em raras ocasiões, use sempre letras minúsculas.

--Feche o parêntese assim que abri-lo.

1. Gravar dados dentro do R (criar *objetos*)

Suponha você queira gravar o valor "4" e chamá-lo de "a":

```
a<-4
```

2. Ver conteúdo de um objeto

Basta digitar o nome do objeto e pressionar a tecla Enter:

```
a
```

```
[1] 4
```

O número entre colchetes apenas indica o número da linha.

3. Operações aritméticas e sequências

Use "*" para multiplicação e "^" para exponenciação:

```
a^2
```

```
[1] 16
```

```
a/2
```

```
[1] 2
```

```
(a^5)/2
```

```
[1] 512
```

```
a<-a^2 #Note que aqui você substitui o valor de "a" que era "4" por "16"
```

```
1:10 #Para fazer sequência de 1 até 10.
```

4. Uso geral:

```
b<-maria(informação1, informação2)
```

O comando acima indica que a função chamada "maria" será aplicada sobre as informações (no R chamadas de argumentos) "informação1" e "informação2" e gravar o resultado dentro de "b". Caso não queira gravar os resultados, basta excluir a parte antes do nome da função. Veja exemplos:

```
a<-sum(3,5)
```

```
a
```

```
log(10)
```

```
log(10,10) # a segunda informação da função log indica a base. Para saber disto, basta digitar ?log
```

```
log(10,2)
```

5. Visualizar arquivo de ajuda:

```
?log
```

```
?sum
```

Veja também no menu as opções de ajuda, incluindo manuais.

6. Importar tabela de dados

Exemplo: Prepare o seguinte conjunto de dados no Excel:

	sp1	sp2	sp3	sp4
ua1	1	2	3	8
ua2	2	3	4	9
ua3	3	4	5	10
ua4	4	5	6	11
ua5	5	6	7	12

Grave o arquivo no formato .txt (texto separado por tabulações) no mesmo diretório que contém o arquivo do R que vocês está usando. Chame o arquivo de cupim.txt

Vamos importa-lo:

```
dados.cupim<-read.table("cupim.txt", header=TRUE)
```

O argumento `header` indica que nossos dados possuem legendas nas linhas e nas colunas.

Caso os dados estejam em outro diretório:

```
dados.cupim<-read.table("C:\\maria\\cupim.txt", header=TRUE)
```

7. Editar dados

```
dados.cupim<-edit(dados.cupim)
```

Abre-se uma janela com nossos dados. Substitua a primeira célula por 10. Note que a nova planilha será gravada como "dados.cupim". Perderemos portanto nossa planilha antiga. Caso queira manter a original, basta gravar com nome diferente.

8. Selecionando linhas e colunas de matrizes [linha, coluna]

```
dados.cupim[1,] # todos elementos da primeira linha
```

```
dados.cupim[,1] # todos elementos da primeira coluna
```

```
dados.cupim[1,1] # o elemento situado na primeira linha e primeira coluna
```

9. Fazer diagrama de dispersão:

```
plot(dados.cupim[,1], dados.cupim[,2])
```

Faz gráfico usando primeira coluna como eixo "x" e segunda coluna como eixo "y".

10. Instalando e carregando pacotes

Uma das coisas mais interessantes no R é a enorme quantidade de "pacotes" disponíveis. Cada pacote inclui um conjunto de funções para determinado tipo de análise. Várias análises multivariadas e de diversidade estão no pacote `vegan` (*vegetation analysis*).

No menu, selecione "Packages" e depois "Install package from CRAN". Selecione o pacote `vegan` na lista que aparecer. Volte novamente no menu "Packages" e selecione "Load packages". Selecione `vegan` na lista que aparecer.

Vamos ver o que este pacote faz. No menu, selecione "Help" e depois "Html help". Clique em "Packages" e depois "vegan".

11. Algumas demonstrações:

Obtenha uma lista dos demos disponíveis:

```
demo( )
```

Vamos ver dois deles:

```
demo(graphics)
```

Veja se a janela do console está ativa e clique "Enter" para ver as várias demonstrações.

```
demo(image)
```

12. Outros comandos úteis:

```
c(1,2,3,a) # concatenar (juntar) objetos
```

```
rep(2,5) #repetição
```

```
seq(1,10,2) #gerar sequência
```

```
max()
```

```
min()
```

```
range()
```

```
sum()
```

```
mean()
```

```
var() # variância
```

```
length() #comprimento (e.g. número de valores de um vetor)
```

```
sort() # ordenar
```

```
rnorm(10) # obtenção de valores aleatórios sorteados de distribuição normal centrada em "0".
```

```
runif(10) # obtenção de valores aleatórios sorteados de distribuição uniforme entre "0" e "1".
```

```
data.frame() #fazer conjuntos de dados
```

```
matrix() ; is.matrix() ; as.matrix()
```

```
matrix[,]
```

```
t(cupim) # transpor matriz
```

```
nrow(cupim) ; ncol() #número linhas e colunas de matriz ou data-frame
```

```
attach(cupim) #anexa conjunto de dados de modo a poder chamar colunas diretamente
```

```
pt() #valor de p da distribuição t
plot(x,y) #fazer diagramas de pontos, linhas etc
hist(x) #fazer histograma
image(as.matrix(cupim)) # entrada com matrix; valores dão a intensidade
image(matrix(1:100,10,10))
ifelse(cupim>0,1,0) #teste lógico;e.g. se valor maior que 0, colocar 1, se não colocar 0.
```

13. Como fazer uma função

Abra o bloco de anotações (*notepad*) e escreva o seguinte:

```
simpson<-function(entrada) {
soma<-sum(entrada)
prop<-entrada/soma
prop2<-prop^2
D<-sum(prop2)
resu<-1-D
return(resu)
}
```

Copie tudo (Ctrl+c) e cole (Ctrl+v) no R. A função (seu programa de computador) esta pronta!!!. Para calcular o índice de diversidade de Simpson, basta você aplicá-la a uma sequência (vetor) de dados, onde cada valor seja a abundância de uma dada espécie.

12. Exemplos de usos

- 1)

```
a<-rnorm(50)
b<-rnorm(50)
plot(a,b)
h<-chull(a,b)
polygon(a[h], b[h], dens=15, angle=30)
```
- 2)

```
a<-1:100
c<-matrix(a,10,10)
image(c)
image(t(c))
image(t(c[,10:1]))
```
- 3)

```
c<-1:100
sample(c,10)
table(sample(c,100))
table(sample(c,100,replace=TRUE))
```
- 4)

```
x<-1:10
y<-(1:10)^2
y<-y+0.5
plot(x,y) ## Deixe a janela do gráfico aberta.
nls(y~x^z,start=list(z=1))
curve(x^z, col=4, add=TRUE) ## substituir valor de 'z' obtido comando anterior
```
- 5)

```
a<-function(x,y){x+y}
x<-seq(-10,10,length=20)
xs<-outer(x^2,x^2,a)
xm<-x%o%x
persp(1740-7*xs-3*xm,phi=25,theta=-35,shade=1,border=4)
```