

PROGRAMA DE PÓS-GRADUAÇÃO EM ECOLOGIA E EVOLUÇÃO - UNIVERSIDADE FEDERAL DE GOIÁS
CURSO: MÉTODOS DE ANÁLISE DE DADOS EM ECOLOGIA DE COMUNIDADES

Prof. Adriano Sanches Melo - asm.adrimelo@naquele@gmail.com
 Dep. Ecologia, ICB, Universidade Federal de Goiás
 www.ecologia.ufrgs.br/~adrimelo/div

AULA 1 – FÓRMULAS NO MS EXCEL E COMANDOS NO R

No Excel:

Fixação de linhas, colunas e células com "\$"
 Funções mais comuns (soma, raiz quadrada, exponenciação)
 Aleatorização usando sequência auxiliar de números aleatórios
 Gravação arquivos em texto

No R

Histórico de origem. Um programa para cálculos e programação. Pacotes
 Diretório de trabalho
 Menu
 Linhas de comando: setas para voltar
 Definição e tipos de objeto: `vector`, `matrix`, `array`, `factor`, `data.frame`, `list`, `function`
 Noções de programação

0. Informações gerais:

--Use ponto "." para casas decimais.
 --Exceto em raras ocasiões, use sempre letras minúsculas.

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<-ufg(informação1, informação 2)
```

O comando acima indica que a função chamada “ufg” 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.
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 diretório da disciplina “C:/div/” com o nome “cupim”.

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 que não aquele onde se encontra o arquivo de dados do R, por exemplo no diretório ‘temp’:

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

Uma outra opção é copiar no Excel e:

```
dados.cupim<-read.table("clipboard", 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
dados.cupim[-c(1,3),] # todas as linhas com exceção das linhas 1 e 3.
```

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. No curso usaremos o pacote `vegan` (análises em ecologia de comunidades).

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. Você não precisará instalar o `vegan` novamente, mas precisará carregá-lo cada vez que iniciar o R.

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
```

```
runif(10) # obtenção de valores aleatórios sorteados de distribuição normal centrada em “0”.
```

```
rnorm(10) # obtenção de valores aleatórios sorteados de distribuição uniforme entre “0” e “1”.
```

```
scan() # importação de valores pelo console (uso de Ctrl V)
```

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

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

```
matrix[,]
```

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

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

```
attach(dados.cupim) #anexa conjunto de dados, podendo chamar colunas diretamente
```

```
pt() #valor de p da distribuição t
```

```
plot(x,y) #fazer diagramas de pontos, linhas etc
```

```
hist(x) #fazer histograma
```

```
image(dados.cupim) #gráfico de superfície; entrada com matrix; valores dão a intensidade
```

```
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 editor de notas (*notepad*)

Escreva o seguinte:

```
ufg<-function(entrada) {  
temp1<-entrada*10
```

```
temp2<-temp1^2
return(temp2)
}
```

Copie tudo e cole no R. A função (seu programa de computador) esta pronta!!!. Note que ela pegará o valor dentro de parênteses e multiplicará por 10. O resultado parcial será então elevado ao quadrado.

```
ufg(entrada=10)
[1] 10000
```

```
ufg(entrada=5)
[1] 2500
```

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, angles=30)
```

```
2) a<-rnorm(50)
   a<-matrix(a,10,5)
   c
   c+2
   c<-c(1:100)
   c<-matrix(c,10,10)
   image(a)
   image(c)
   image(t(c))
   image(t(c[10:1]))
   image(t(c[10:1,10:1]))
```

```
3) sample(c,10)
   amostra<-sample(c,100)
   table(amostra)
   amostra<-sample(c,100, replace=TRUE)
   table(amostra)
```

```
4) x<-1:10
   y<-ufg(c(1:10))
   plot(x,y)
   nls(y~x^z, start=list(z=1)) ## se aparecer mensagem de erro, tente outros
                               valores de z
   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)
```