

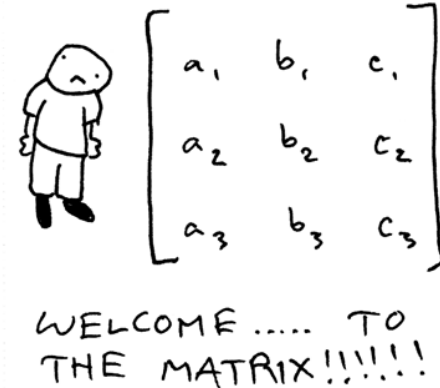


Curso de introdução ao software R - Modulo 1

Darlene Duarte

Renata Veroneze

Matrizes



- Operações com matrizes:

```
>m1<-matrix(data=1:9,nrow=3,ncol=3)
```

```
>m2<-matrix(data=10*(1:9),nrow=3,ncol=3)
```

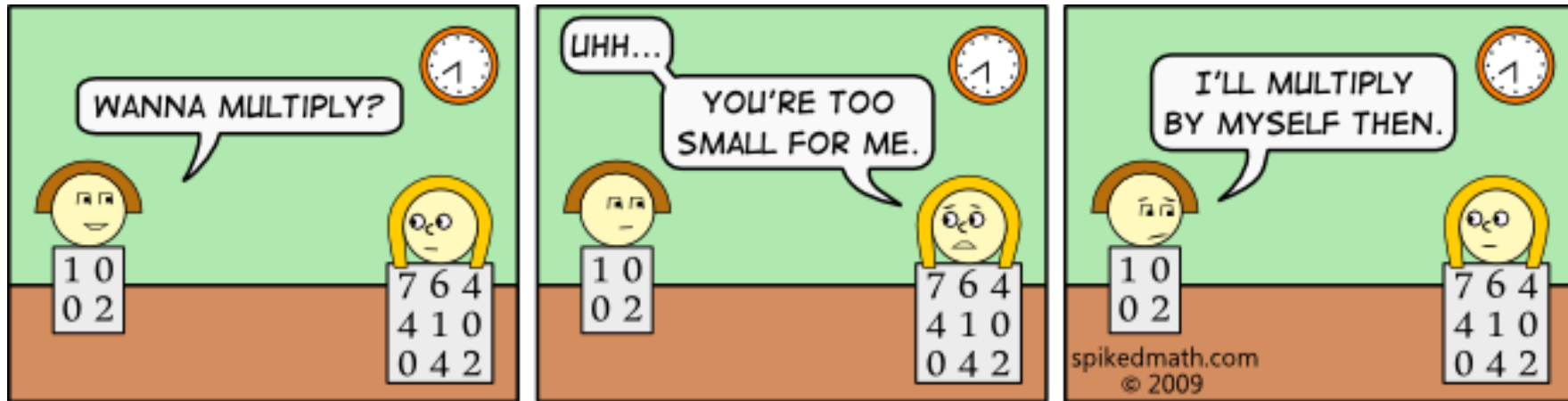
```
>m3<-m1+m2 #somando elementos das matrizes
```

```
>m4<-m1-m2 #subtraindo elementos das matrizes
```

```
>m5<-m1*m2 #multiplica cada elemento da matriz
```



Matrizes



- Multiplicação de matriz:
`>m6<-m1%*%m2`



Matrices

- Transpor matriz:

```
>m7<-t(m6)
```

matrix	it's transpose																		
<table><tr><td>a</td><td>c</td><td>b</td></tr><tr><td>b</td><td>a</td><td>c</td></tr><tr><td>c</td><td>b</td><td>a</td></tr></table>	a	c	b	b	a	c	c	b	a	<table><tr><td>a</td><td>b</td><td>c</td></tr><tr><td>c</td><td>a</td><td>b</td></tr><tr><td>b</td><td>c</td><td>a</td></tr></table>	a	b	c	c	a	b	b	c	a
a	c	b																	
b	a	c																	
c	b	a																	
a	b	c																	
c	a	b																	
b	c	a																	



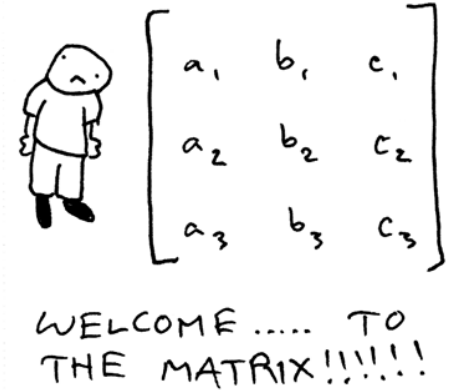
Matrizes

- Inversa

```
>m8<-solve(m7) #Fail
```

```
library(MASS)
```

```
>m8<-ginv(m7)
```



Desafio



1. Crie uma matriz X (3x3)
2. Imprima a matriz na tela usando a função `print`
3. Obtenha a inversa da matriz X
4. Imprima a matriz X inversa na tela usando a função `print`
5. Salve essas funções em um script
6. Execute o script salvo utilizando o comando `source`



Desafio



1. Dado:

X	1	2	3	4	5	6	7	8	9	10
y	13.4	16.6	15.8	17.3	18.5	22.1	23.2	35.9	31.3	39.4

Sendo: $\beta = (X'X)^{-1} X'y$

Obtenha β



Tipos de objetos

- **Vetores:** uma sequência de valores numéricos ou de caracteres (letras, palavras).
- **Matrizes:** coleção de vetores em linhas e colunas.
- **Dataframe:** O mesmo que uma matriz, mas aceita vetores de tipos diferentes (numérico e caracteres).
- **Listas:** conjunto de vetores, dataframes ou de matrizes.
- **Funções**



Lista

```
lista <- list( c(1:5),  
              c("homem", "mulher"),  
              c(T, F, T),  
              list(c(1,2,3), c("a", "b", "c")) )
```

lista



Utilizando dados de busca no google

```
install.packages("gtrendsR")  
library(gtrendsR) # interface com o Google Trends  
  
busca <- c("Zootecnia", "Engenharia  
Florestal", "Engenharia Agricola", "Agronomia",  
"ciências econômicas")  
res <- gtrends(keyword = busca, geo = "BR", time =  
"all")  
head(res)  
plot(res)
```



Lendo arquivos

- Lendo arquivo .txt



```
>ex01 <- read.table("Exemplo01.txt", h=T)
```

```
>head(ex01)
```

```
>tail(ex01)
```



Lendo arquivos



- Lendo arquivo .csv , separado por “:”

```
>ex02 <- read.table("Exemplo02.csv", h=T,  
  sep=":", dec=",")
```

```
>ex02
```



Lendo arquivos



- Lendo arquivo .csv , separado por virgula

```
>ex03 <- read.csv("Exemplo01.csv", h=T)
```

```
>head(ex03)
```



Visualizando partes do curso

Visualizando partes do arquivo

```
>ex03[1:10,]
```

```
>ex03[1:10,2:3]
```

```
>ex03$tratamento
```



Data frame com partes dos dados

Criando um data frame com parte dos dados

```
>data_novo<-ex03[1:10,2:3]
```

```
>data_novo
```



Verificando número de níveis

Verificando número de níveis

```
>trat<-ex03$tratamento
```

```
>trat_unico<-unique(trat)
```

```
>trat_unico
```

#ou

```
>trat_unico<-factor(trat)
```



Ordenando bancos de dados

Ordenando banco de dados de acordo com a coluna desejada:

```
>ord_resp<-ex03[order(ex03[,3]),]  
#ordenando pela coluna 3
```

```
>ord_resp
```



Extraindo parte do arquivo

```
>ex03[order(ex03$resposta),]
```

Extraindo parte do arquivo:

```
>subset(ex03, ex03$tratamento==1)
```

```
>subset(ex03, ex03$tratamento > 2)
```



Modo mais elegante

```
d<-seq(1,2)
```

```
ex04<-ex03[ex03$tratamento %in% d,]
```

Adicionando informações

Adicionando mais linhas ou colunas a um arquivo usando as funções *rbind()* e *cbind()*

```
resp<-ex03$resposta
```

```
r2<-resp^2
```

```
novo<-cbind(ex03, r2)
```

Exportando arquivos

Se desejarmos salvar um arquivo em nossa pasta?

Existem algumas funções para isso

`write.csv` #usa “.” como separador decimal e “,” como separador de coluna

`write.csv2` #usa “,” como separador decimal e “;” como separador de coluna

`write.table` #cria um arquivo txt

`write` #cria um arquivo sem extensão



Exportando arquivos

Como exemplo vamos exportar o arquivo que geramos ordenando o banco de dados (ord_resp):

```
>write.table(ord_resp, "ordenado.txt",  
col.names=T,row.names=F, quote=F,sep="\t")
```

Desafio



Leia o arquivo chamado `exercicio3.5.txt`, selecione os dados da região "Bofete" e separe em um novo objeto denominado *bofete*. Salve-o em um arquivo denominado *bofete.txt* no diretório corrente, separado por tabulação e sem nome de linhas.



Resposta

```
> exe<-read.table("exercicio3.5.txt",h=T,sep=";")
```

```
>bofete<-subset(exe, exe$regiao=="Bofete")
```

```
write.table(bofete, "bofete.txt",  
col.names=T,row.names=F, quote=F,sep="\t")
```



Desafio



Crie um novo objeto chamado "dados.esaligna2" incluindo aos dados do arquivo-texto `exercicio3.5_2.txt` uma nova coluna que contém a soma das biomassas de folhas e do tronco de cada árvore após as outras colunas.



Resposta

```
> dados.esaligna<-  
read.table("exercicio3.5_2.txt", h=T, sep=",")
```

```
> dados.esaligna$biomassa<-  
dados.esaligna$folha+dados.esaligna$tronco
```



Desafio



- Crie um novo objeto chamado "dados.esaligna.10cm", selecionando apenas os dados relativos a árvores com mais de 10 cm de diâmetro na altura do peito (dap) do objeto dados.esaligna2.
- Salve os objetos criados em arquivos txt, separados por virgula.



Resposta

```
> dados.esaligna.10cm<-  
subset(dados.esaligna2,dados.esaligna2$dap  
>10)
```

```
> head(dados.esaligna.10cm)
```

```
>write.table(dados.esaligna.10cm, "  
dados.esaligna.10cm.txt",  
col.names=T,row.names=F, quote=F,sep=",")
```

