

BIE5782

Unidade 4:

**OBJETOS DE DADOS
E SUA MANIPULAÇÃO**

Como Importar Dados para o R?

	A	B	C	D
1		nascimento	estado	vivo
2	Didi	1936	CE	s
3	<u>Dedé</u>	1936	RJ	s
4	<u>Mussum</u>	1941	RJ	n
5	Zacarias	1934	MG	n



Salvar em arquivo-texto (csv, txt)

```
;"nascimento";"estado";"vivo"  
"Didi";1936;"CE";"s"  
"Dedé";1936;"RJ";"s"  
"Mussum";1941;"RJ";"n"  
"Zacarias";1934;"MG";"n"
```

read.table

Leitura de Arquivos-texto

```
> trapalhoes <- read.table("trapalhoes.csv",  
+ header=T, sep=";", row.names=1)
```

```
> trapalhoes
```

	nascimento	estado	vivo
Didi	1936	CE	s
Dedé	1936	RJ	s
Mussum	1941	RJ	n
Zacarias	1934	MG	n

Para os argumentos de “read.table” consulte a ajuda. Faça o mesmo para para “write.table”.

names, rownames

Nomes de “Colunas” e de Linhas

```
> names(trapalhoes)
[1] "nascimento" "estado"      "vivo"
> names(trapalhoes) <- c("NASC", "ESTADO",
+ "VIVO")
> trapalhoes
      NASC ESTADO  VIVO
Didi   1936    CE     s
Dedé   1936    RJ     s
Mussum 1941    RJ     n
Zacarias 1934   MG     n
> rownames(trapalhoes)
[1] "Didi"      "Dedé"      "Mussum"
[4] "Zacarias"
```



Seleção de “Colunas”

```
> trapalhoes$nascimento  
[1] 1936 1936 1941 1934
```

```
> trapalhoes$vivo  
[1] s s n n  
Levels: n s
```

```
> trapalhoes$estado  
[1] CE RJ RJ MG  
Levels: CE MG RJ
```

\$ com atribuição

Substituição e Criação de “Colunas”

```
> trapalhoes$VIVO <- c("T", "T", "F", "F")
```

```
> trapalhoes$ID.2008 =  
+ 2008 - trapalhoes$NASC
```

```
> trapalhoes
```

	NASC	ESTADO	VIVO	ID.2008
Didi	1936	CE	T	72
Dedé	1936	RJ	T	72
Mussum	1941	RJ	F	67
Zacarias	1934	MG	F	74

class

Classes de Objetos

```
> class(trapalhoes)
[1] "data.frame"
> class(trapalhoes$NASC)
[1] "integer"
> class(trapalhoes$VIVO)
[1] "character"
> class(trapalhoes$ESTADO)
[1] "factor"
```

O objeto “trapalhoes” é um objeto da classe “data.frame”, que são conjuntos de vetores de mesmo comprimento, mas que podem ser de classes diferentes.

list

Cria Objeto da Classe “Lista”

```
> a
[1] 1 2 3 4 5
> b
[1] a a a b b b c c c
Levels: a b c
> c
  sec inicio
1 XIX    1801
2  XX    1901
3 XXI    2001

> minha.lista <- list(um.vetor=a, um.fator=b,
+ um.data.frame=c)
```

Listas (cont.)

```
> minha.lista
```

```
$um.vetor
```

```
[1] 1 2 3 4 5
```

```
$um.fator
```

```
[1] a a a b b b c c c
```

```
Levels: a b c
```

```
$um.data.frame
```

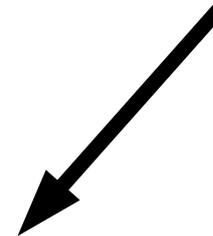
```
  sec inicio
```

```
1 XIX  1801
```

```
2  XX  1901
```

```
3 XXI  2001
```

Uma lista pode conter
outras listas
(**recursividade!**)



```
> minha.lista$um.data.frame$inicio
```

```
[1] 1801 1901 2001
```

`as.character`, `as.logical` ...

Coerção de Classes de Objetos

```
> class(trapalhoes$VIVO)
[1] "character"
> trapalhoes$VIVO
[1] "T" "T" "F" "F"
> trapalhoes$VIVO <-
+ as.logical(trapalhoes$VIVO)
> trapalhoes$VIVO
[1] TRUE TRUE FALSE FALSE
> class(trapalhoes$VIVO)
[1] "logical"
```

Para cada classe há uma função de coerção, sempre com a forma `as.classe`.

factor

Classe de fatores

```
> sexo <- rep(c("F", "M"), each=9)
> sexo
[1] "F" "F" "F" "F" "F" "F" "F" "F" "F" "M"
[11] "M" "M" "M" "M" "M" "M" "M" "M" "M"
> class(sexo)
[1] "character"

> sexo <- factor(rep(c("F", "M"), each=9))
> sexo
 [1] F F F F F F F F F M M M M M M M M M
Levels: F M
> class(sexo)
[1] "factor"
```

table

Contagens de Observações

```
> sexo
[1] F F F F F F F F F M M M M M M M M
Levels: F M
> table(sexo)
sexo
F M
9 9
> dieta
[1] L L L D D D N N N L L L D D D N N N
Levels: N D L
> table(sexo,dieta)
      dieta
sexo  N  D  L
  F   3  3  3
  M   3  3  3
```

tapply

Operações por níveis dos Fatores

```
> sexo
```

```
[1] F F F F F F F F F M M M M M M M M M
```

```
Levels: F M
```

```
> peso <-
```

```
c(66,70,67,79,82,54,61,59,63,85,78,112,92,77,82,  
+ 85,80,91)
```

```
> tapply(X=peso, INDEX=sexo, FUN=mean)
```

```
      F      M
```

```
66.77778 86.88889
```

```
> tapply(X=peso, INDEX=sexo, FUN=range)
```

```
$F
```

```
[1] 54 82
```

```
$M
```

```
[1] 77 112
```

> , < , == , != , & , | , ...

Operadores Lógicos

```
> ALTURA
```

```
[1] 1.85 1.78 1.92 1.63 1.81 1.55
```

```
> SEXO
```

```
[1] M M M F F F
```

```
Levels: F M
```

```
> ALTURA > 1.80
```

```
[1] TRUE FALSE TRUE FALSE TRUE FALSE
```

```
> homens.altos <- ALTURA >1.80 & SEXO == "M"
```

```
> homens.altos
```

```
[1] TRUE FALSE TRUE FALSE FALSE FALSE
```

```
> class(homens.altos)
```

```
[1] "logical"
```

Soma de Vetores Lógicos?

```
> notas.dos.alunos
```

```
[1] 6.0 5.1 6.8 2.8 6.1 9.0 4.3 10.4  
[9] 6.0 7.9 8.9 6.8 9.8 4.6 11.3 8.0  
[17] 6.7 4.5
```

```
##Quantos aprovados?
```

```
> sum(notas.dos.alunos>=5)
```

```
[1] 14
```

```
##Qual a proporção de aprovados?
```

```
> sum(notas.dos.alunos>=5)
```

```
+ /length(notas.dos.alunos)
```

```
[1] 0.7777778
```

Um Exemplo: Totais Marginais

```
> arqu
  a  b  c  d  e  f  g  h
1  2 10 8  0 10 10 10  0
2 10  9 0  7  8  0  7  0
3  8  0 0  8 11  0  8  0
4 11 10 8  9  3  9  8  7
5  3  0 0  5  9  0  6  0
```

```
> arqu.bin <- arqu > 0
> arqu.bin
```

```
  a      b      c      d      e      f      g      h
1 TRUE  TRUE  TRUE FALSE TRUE  TRUE TRUE  FALSE
2 TRUE  TRUE FALSE  TRUE TRUE  FALSE TRUE  FALSE
3 TRUE FALSE FALSE  TRUE TRUE  FALSE TRUE  FALSE
4 TRUE  TRUE  TRUE  TRUE TRUE  TRUE  TRUE  TRUE
5 TRUE FALSE FALSE  TRUE TRUE  FALSE TRUE  FALSE
```

apply

Totais Marginais (cont.)

```
> arqu.bin
```

	a	b	c	d	e	f	g	h
1	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE
2	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE
3	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE
4	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
5	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE

```
> ## Quantas espécies por ilha?
```

```
> apply(arqu.bin, MARGIN=2, FUN=sum)
```

a	b	c	d	e	f	g	h
5	3	2	4	5	2	5	1

```
> ## Quantas ilha por espécie?
```

```
> apply(arqu.bin, MARGIN=1, FUN=sum)
```

1	2	3	4	5
6	5	4	8	4

E lá vem MAIS história!

SUBCONJUNTOS E INDEXAÇÃO

Operador [

Indexação de Vetores

```
> x
[1] "A" "B" "C" "D" "E" "F"
> x[1]
[1] "A"
> x[length(x)]
[1] "F"
> x[1:3]
[1] "A" "B" "C"
> x[c(1,3,5)]
[1] "A" "C" "E"
> x[-2]
[1] "A" "C" "D" "E" "F"
> x[-c(2,4)]
[1] "A" "C" "E" "F"
```

Indexação com Operações Lógicas

```
> ALTURA
```

```
[1] 1.85 1.78 1.92 1.63 1.81 1.55
```

```
> PESO
```

```
[1] 80 100 115 70 65 50
```

```
> SEXO
```

```
[1] M M M F F F
```

```
Levels: F M
```

```
> homens.altos <- ALTURA >1.80 & SEXO ==  
"M"
```

```
> homens.altos
```

```
[1] TRUE FALSE TRUE FALSE FALSE FALSE
```

```
> PESO[homens.altos]
```

```
[1] 80 115
```

```
> PESO[ALTURA >1.80 & SEXO == "M"]
```

```
[1] 80 115
```

[com atribuição

Alteração de Subconjuntos

```
> trapalhoes
```

	NASC	ESTADO	VIVO	ID.2008
Didi	1936	CE	TRUE	72
did	1936	RJ	TRUE	72
Mussum	1941	RJ	TRUE	67
Zacarias	1934	MG	FALSE	74

```
> trapalhoes$VIVO[3] <- F
```

```
> rownames(trapalhoes)[2] <- "Dedé"
```

```
> trapalhoes
```

	NASC	ESTADO	VIVO	ID.2008
Didi	1936	CE	TRUE	72
Dedé	1936	RJ	TRUE	72
Mussum	1941	RJ	FALSE	67
Zacarias	1934	MG	FALSE	74

Indexação em mais de uma Dimensão

```
> trapalhoes
```

```
      NASC ESTADO  VIVO ID.2008
Didi   1936     CE  TRUE      72
Dedé   1936     RJ  TRUE      72
Mussum 1941     RJ FALSE     67
Zacarias 1934    MG FALSE     74
```

```
> trapalhoes[3,3]
```

```
[1] "FALSE"
```

```
> trapalhoes[3,]
```

```
      NASC ESTADO  VIVO ID.2008
Mussum 1941     RJ FALSE     67
```

```
> trapalhoes[,3]
```

```
[1] "TRUE"  "TRUE"  "FALSE" "FALSE"
```

```
> trapalhoes[2,c(1,3)]
```

```
      NASC  VIVO
Dedé   1936  TRUE
```

matrix

Cria Objeto da Classe “Matriz”

```
> matrix(1:12, nrow=4, ncol=3)
```

```
      [,1] [,2] [,3]  
[1,]    1    5    9  
[2,]    2    6   10  
[3,]    3    7   11  
[4,]    4    8   12
```

```
> matrix(1:12, 4, 3, byrow=T)
```

```
      [,1] [,2] [,3]  
[1,]    1    2    3  
[2,]    4    5    6  
[3,]    7    8    9  
[4,]   10   11   12
```

matrix

Cria Objeto da Classe “Matriz”

```
> m1 <- matrix(letters[1:12],3,4)
> m1
      [,1] [,2] [,3] [,4]
[1,] "a"  "d"  "g"  "j"
[2,] "b"  "e"  "h"  "k"
[3,] "c"  "f"  "i"  "l"
> rownames(m1) <- paste("linha",1:3,sep="")
> colnames(m1) <- paste("coluna",1:4,sep="")
> m1
      coluna1 coluna2 coluna3 coluna4
linha1 "a"      "d"      "g"      "j"
linha2 "b"      "e"      "h"      "k"
linha3 "c"      "f"      "i"      "l"
```

Permutação por Indexação

```
> zoo
onça anta tatu paca
      4    10     2    45
> zoo[4:1]
paca tatu anta onça
     45     2    10     4
> zoo[c(3,4,1,2)]
tatu paca onça anta
     2    45     4    10
```

order

Ordenação por Indexação

```
> indice <- order(names(zoo))
> indice
[1] 2 1 4 3
> zoo[indice]
anta onça paca tatu
   10    4   45    2
> indice <-
order(names(zoo), decreasing=T)
> indice
[1] 3 4 1 2
> zoo[indice]
tatu paca onça anta
   2   45    4   10
```

order

Ordenação por Múltiplos Critérios

```
> cidades[order(cidades$regiao, cidades$estado,  
+ cidades$pop.2007, decreasing=T), ]
```

	regiao	estado	pop.2007
São Paulo	SE	SP	11104712
Guarulhos	SE	SP	1289047
Campinas	SE	SP	1073020
Osasco	SE	SP	724368
Santo André	SE	SP	676846
Sorocaba	SE	SP	590846
Rio de Janeiro	SE	RJ	6178762
Recife	NE	PE	1528970
Jaboatão	NE	PE	661901
Salvador	NE	BA	2714119
Feira de Santana	NE	BA	544113

FIM DA UNIDADE 4

Para amanhã:

Leia o texto do Wiki e
faça os exercícios 4.1 a 4.18

<http://cmq.esalq.usp.br/wiki/doku.php?id=biometria:r-tutor:04-dados>

IMPORTANTE: O texto do Wiki tem outras informações necessárias para os exercícios, como funções e classes de objetos que não foram detalhadas nesta apresentação.