

# Introdução ao R

Lupércio F. Bessegato & Ronaldo Bastos & Marcel T. Vieira

UFJF – Departamento de Estatística  
2010



---

---

---

---

---

---

---

---

## Roteiro

1. Informações Gerais
2. Aritmética
3. Variáveis e Vetores
4. Scripts
5. Matrizes e Arrays

---

---

---

---

---

---

---

---

## Introdução

- Ferramenta de análise estatística de dados com muitos recursos e possibilidades;
- Constantemente atualizado e revisado incluindo novas técnicas e métodos;
- Programa de domínio público:  
√ <http://cran.r-project.org>

---

---

---

---

---

---

---

---

## Download do Programa

The screenshot shows the 'Download and Install R' page. Annotations include: an arrow pointing to the 'Linux', 'MacOS X', and 'Windows' list; an arrow pointing to the 'Download R 2.11.1 for Windows (32 bits)' link; an arrow pointing to the 'base' subdirectory; and an arrow pointing to the 'binaries' section. The page text includes: 'Download and Install R', 'Precompiled binary distributions of the base system and contributed packages, Windows and Mac users most likely want one of these versions of R:', 'Source Code for all Platforms', 'This directory contains 32-bit binaries: See here for a 64-bit Windows port.', 'Note: CRAN does not have Windows', 'Subdirectories: base, contrib', 'Binaries! Binaries!', 'Please do not submit binaries to CRAN to Windows binaries.', 'Frequently asked questions', and 'Please see the R FAQ for general information about R and the R Windows FAQ for Windows-specific s'.

---

---

---

---

---

---

---

---

---

---

## Comandos

- Para solicitar uma tarefa do R podemos digitar uma linha de comando no console

```
R Console
R version 2.11.0 (2010-04-22)
Copyright (C) 2010 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R é um software livre e vem sem GARANTIA ALGUMA.
Você pode redistribuí-lo sob certas circunstâncias.
Digite 'license()' ou 'licence()' para detalhes de distribuição.

R é um projeto colaborativo com muitos contribuidores.
Digite 'contributors()' para obter mais informações e
'citation()' para saber como citar o R ou pacotes do R em publica

Digite 'demo()' para demonstrações, 'help()' para o sistema on-li
ou 'help.start()' para abrir o sistema de ajuda em HTML no seu na
Digite 'q()' para sair do R.

> sqrt(4)
[1] 2
> |
```

---

---

---

---

---

---

---

---

---

---

- Todas as funções do R devem ser digitadas em letras **minúsculas**  
√ O R é sensível a letras maiúsculas e minúsculas.
- Todas as palavras-chaves do R estão em letras minúsculas

---

---

---

---

---

---

---

---

---

---

### Alguns Comandos

- Soma → +
- Subtração → -
- Divisão → /
- Multiplicação → \*
- Potenciação → \*\* ou ^
- Raiz quadrada de n → sqrt(n)
- Logaritmo natural → log(n)

---

---

---

---

---

---

---

---

### Help

- Help no R é bastante completo e auto-suficiente.
- Se não conhecemos o comando de alguma técnica podemos chamar a função *help*:
  - √ help(palavra-chave)

---

---

---

---

---

---

---

---

### Help na Web

- R Project
  - √ <http://www.r-project.org>
- Tutorial de Introdução ao R
  - √ <http://leg.ufpr.br/Rtutorial/contents.html>
- R Graphical Manual
  - √ <http://bg9.imslab.co.jp/Rhelp>

---

---

---

---

---

---

---

---

### Comando de Atribuição – Variável

- Atribuição de valor à variável  $x$  (ou qualquer outro 'nome')  
√  $x < 4$  ou  $x = 4$
- Atribuição de valores às variáveis  $x1$  e  $x2$ :  
√  $x1 = -4$   
√  $x2 = 6$
- Atribua à variável  $z$  o valor da soma de  $x1$  e  $x2$
- Evitar usar acentos ou ç na denominação das variáveis

---

---

---

---

---

---

---

---

### Operações com as Variáveis

- Calcular:

$$x^{x1} + \frac{z}{x2}$$

```
> x^x1+z/x2  
[1] 2.916667
```

---

---

---

---

---

---

---

---

### Apagar Variável

- Comando `rm(variavel)`  
√ Cuidado! Não tem undo  
√ `rm(x,x1,x2,z)` ou `rm(x); rm(x1); rm(x2); rm(z)`

---

---

---

---

---

---

---

---

## Vetores Numéricos

- Comando concatenar: c(dados)

√ Criar o vetor peso com os valores 60, 72, 57, 90, 95, 72

```
> peso<- c(60, 72, 57, 90, 95, 72)
> peso
[1] 60 72 57 90 95 72
>|
```

√ Criar o vetor altura com os valores 1,75; 1,80; 1,65; 1,90; 1,74; 1,91

```
> altura= c(1.75, 1.80, 1.65, 1.90, 1.74, 1.91)
> altura
[1] 1.75 1.80 1.65 1.90 1.74 1.91
>|
```

---

---

---

---

---

---

---

---

## Aritmética de Vetores

- O R pode manipular vetores como simples objetos
- Cálculo do Índice de Massa Corporal

$$IMC = \frac{peso}{altura^2}$$

```
>
> imc<-peso/altura^2
> imc
[1] 19.59184 22.22222 20.93664 24.93075 31.37799 19.73630
>|
```

---

---

---

---

---

---

---

---

## Cálculo da Média de Peso

- Média aritmética

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

```
> sum(peso)
[1] 446
> length(peso)
[1] 6
>
> xbarra<-sum(peso)/length(peso)
> xbarra
[1] 74.33333
>
```

---

---

---

---

---

---

---

---

- Podemos digitar uma seqüência de comandos em um editor de textos (p.ex. o Bloco de Notas, e depois colar no console para execução das tarefas solicitadas

```
sum(peso); length(peso); xbarra=sum(peso)/length(peso)
```

---

---

---

---

---

---

---

---

- Cálculo de

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

```
> peso - xbarra
[1] -14.333333 -2.333333 -17.333333 15.666667 20.666667 -2.333333
>
> (peso - xbarra)^2
[1] 205.444444 5.444444 300.444444 245.444444 427.111111 5.444444
>
> sum(peso - xbarra)^2
[1] 1189.333
>
> sqrt(sum((peso - xbarra)^2)/(length(peso)-1))
[1] 15.42293
> .
```

---

---

---

---

---

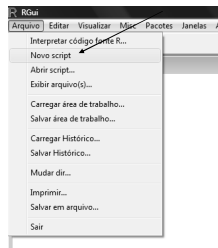
---

---

---

### Uso de Script

- Facilita para:
  - ✓ Correção ou expansão de comandos
  - ✓ Repetição de comandos
  - ✓ Armazenamento de resultados




---

---

---

---


---

---

---

---

## Compilação do Script



```
peso = xbarra
peso = xbarra^2
sum(peso = xbarra^2)
sqrt(sum(peso = xbarra^2)/(length(peso)-1))
```

- Usa-se a tecla F5 para compilar:
  - √ A linha em que se encontra o cursor (no script)
  - √ As linhas selecionadas (no script)
- Resultado compilação no console

---

---

---

---

---

---

---

---

## Vantagens

- Facilidade para corrigir os comandos ou valores
- Possibilidade de armazenar todos os resultados
- Repetição dos passos corretos de toda a sessão de trabalho
- Trabalho 'limpo'
  - √ Para limpar o console usa-se CTRL + L

---

---

---

---

---

---

---

---

## Novos Comandos

- Use os comandos:
  - √ mean(peso)
  - √ sd(peso)
- Compare com os resultados obtidos anteriormente

---

---

---

---

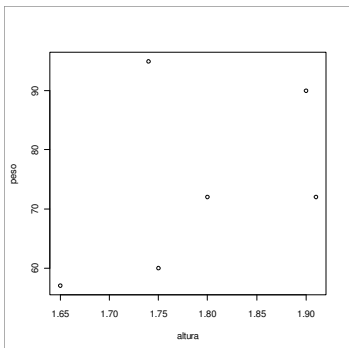
---

---

---

---

- Gráfico de altura vs peso  
 $\sqrt{\text{plot}(\text{altura}, \text{peso})}$




---

---

---

---

---

---

---

---

### Vetores de Caracteres

- Vetor de *strings* de texto  
 $\sqrt{\text{Valores especificados entre aspas (simples ou dupla)}}$
- Exemplo:

```
>
> nomes<-c('Amanda', 'Andréa', 'Camila', 'Carolina', 'Catia', 'Daniele')
> nomes
[1] "Amanda" "Andréa" "Camila" "Carolina" "Catia" "Daniele"
> |
```

---

---

---

---

---

---

---

---

### Vetores Lógicos

- Podem tomar os valores True ou False (ou NA)  
 $\sqrt{\text{c(T,T,F,T)}}$
- Expressões de Relação

```
>
> 1m<25
[1] FALSE FALSE FALSE FALSE TRUE FALSE
>
> |
```

---

---

---

---

---

---

---

---

## Operadores Lógicos

Símbolo	Função
<	Menor que
>	Maior que
<=	Menor que ou igual a
>=	Maior que ou igual a
= =	Igual a
! =	Não igual a
&	E (para combinar expressões)
	Ou (para combinar expressões)
!	Não (para combinar expressões)

---

---

---

---

---

---

---

---

## Funções que Criam Vetores

- Comando **c(valores)**
- Comando **seq**  
√ Usado em séries de números equidistantes

```
>  
> seq(4,9)  
[1] 4 5 6 7 8 9  
>  
> seq(4,10,2)  
[1] 4 6 8 10  
>  
> seq(4,9,0.5)  
[1] 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0  
>  
> 4:9  
[1] 4 5 6 7 8 9
```

---

---

---

---

---

---

---

---

- Comando **rep**  
√ Usado para gerar valores repetidos

```
R Console  
> denovo<-c(7,9,13)  
>  
> rep(denovo,3)  
[1] 7 9 13 7 9 13 7 9 13  
>  
>  
> rep(1:2,c(4,8))  
[1] 1 1 1 1 2 2 2 2 2 2 2 2  
>  
> |
```

---

---

---

---

---

---

---

---

## Matrizes e Arrays

- Matriz:
  - √ Números
  - √ Strings de caracteres (pelo menos um caracter)
- Podem ser representadas como composição de vetores-linha ou vetores-coluna

---

---

---

---

---

---

---

---

- Criação pelo comando *matrix*:

```
> x<-matrix(1:12,nrow=3,byrow=T)
> x
     [,1] [,2] [,3] [,4]
[1,]  1   2   3   4
[2,]  5   6   7   8
[3,]  9  10  11  12
```

- Transposição:

```
> x<-matrix(1:12,nrow=3,byrow=F)
> x
     [,1] [,2] [,3] [,4]
[1,]  1   4   7  10
[2,]  2   5   8  11
[3,]  3   6   9  12
> |
```

```
> t(x)
     [,1] [,2] [,3]
[1,]  1   2   3
[2,]  4   5   6
[3,]  7   8   9
[4,] 10  11  12
> |
```

---

---

---

---

---

---

---

---

- Atribuindo nomes às linhas e colunas

```
> x<-matrix(1:12,nrow=3,byrow=F)
> rownames(x)<-c('Primeira','Segunda','Terceira')
> colnames(x)<-LETTERS[1:4]
> x
      A B C D
Primeira 1 4 7 10
Segunda  2 5 8 11
Terceira 3 6 9 12
> |
```

---

---

---

---

---

---

---

---

- Comandos cbind (ou rbind)  
 √ 'junção' de vetores coluna (ou linhas)

```

> cbind(A=1:4,B=5:8,C=9:12)
  A B C
[1,] 1 5 9
[2,] 2 6 10
[3,] 3 7 11
[4,] 4 8 12
>
> |
>
> rbind(A=1:4,B=5:8,C=9:12)
  [,1] [,2] [,3] [,4]
A     1     2     3     4
B     5     6     7     8
C     9    10    11    12
. !

```

---

---

---

---

---

---

---

---

---

---

### Data Frame

- Lista de vetores e/ou fatores de mesmo comprimento
- Em Estatística:
  - √ Matriz de dados (Data matrix)
  - √ Conjunto de dados (Data set)

---

---

---

---

---

---

---

---

---

---

### Exemplo – Consumo de Energia

- Consumo de energia pré-menstrual:
  - √ consumo.pre = c(5260, 5470, 5640, 6180, 6390)
- Consumo de energia pós-menstrual:
  - √ consumo.pos = c(3910, 4220, 3885, 5160, 5645)
- Matriz de dados (Data frame):
  - √ dados.altman=data.frame(consumo.pre,consumo.p  
os)

---

---

---

---

---

---

---

---

---

---

```

> consumo.pre = c(5260, 5470, 5640, 6180, 6390)
> consumo.pos = c(3910, 4220, 3885, 5160, 5645)
>
> dados.altman = data.frame(consumo.pre, consumo.pos)
>
> dados.altman
  consumo.pre consumo.pos
1      5260      3910
2      5470      4220
3      5640      3885
4      6180      5160
5      6390      5645
> |

```

- Dados são emparelhados
- Variáveis acessíveis usando notação \$  
 $\sqrt{\text{dados.altman}\$consumo.pre}$

---

---

---

---

---

---

---

---

### Indexação

- Consumo pré-menstrual mulher #5  
 $\sqrt{\text{consumo.pre}[5]}$
- Consumo pré-menstrual mulheres #3, 5, 7  
 $\sqrt{\text{consumo.pre}[c(3, 5, 7)]}$
- Consumo pré-menstrual 5 primeiras mulheres  
 $\sqrt{\text{consumo.pre}[1:3]}$

---

---

---

---

---

---

---

---

- Consumo pré-menstrual todas exceto mulheres #3, 5 e 7  
 $\sqrt{\text{consumo.pre}[-c(1, 3, 5)]}$

```

> consumo.pre[5]
[1] 6390
>
> consumo.pre[c(3,5,7)]
[1] 5640 6390 NA
>
>
> consumo.pre[1:3]
[1] 5260 5470 5640
>
> consumo.pre[-c(1,3,5)]
[1] 5470 6180
>
> impar=c(1,3,5)
> consumo.pre[-impar]
[1] 5470 6180
>
> dados.altman$consumo.pre[5]
[1] 6390
> |

```

---

---

---

---

---

---

---

---

- Pode-se indexar pelo do Data frame  
√ dados.altman\$consumo.pre[5]
- Todas as medidas da mulher #5:  
√ dados.altman[5,]
- Consumos de energia pós-menstrual 3 primeiras mulheres  
√ dados.altman[1:3,2]

---

---

---

---

---

---

---

---

```
> dados.altman$consumo.pre[5]
[1] 6390
>
> dados.altman[5,]
consumo.pre consumo.pos
5          6390          5645
>
>
> dados.altman[1:3,2]
[1] 3910 4220 3885
>
> |
```

---

---

---

---

---

---

---

---

### Seleção Condicional

```
√ consumo.pos[consumo.pre>6000]

√ consumo.pos[consumo.pre>6000 & consumo.pre<=6500]

√ consumo.pre>6000 & consumo.pre<=6500]
```

---

---

---

---

---

---

---

---

```

>
> consumo.pos[consumo.pre>6000]
[1] 5160 5645
>
>
> consumo.pos[consumo.pre>6000 & consumo.pos <=6500]
[1] 5160 5645
>
>
> consumo.pre>6000 & consumo.pos <=6500
[1] FALSE FALSE FALSE TRUE TRUE
>
> condicao=consumo.pre>6000 & consumo.pos <=6500
>
> consumo.pos[condicao]
[1] 5160 5645
>
> |

```

---

---

---

---

---

---

---

---

### Ordenação

- Comando sort: ordena os elementos do vetor

√ Default é crescente

√ `sort(consumo.pos)`

- Ordenação decrescente

√ `sort(consumo.pos, decreasing=TRUE)`

```

>
> consumo.pos
[1] 3910 4220 3885 5160 5645
>
>
> sort(consumo.pos)
[1] 3885 3910 4220 5160 5645
>
>
> sort(consumo.pos, decreasing=TRUE)
[1] 5645 5160 4220 3910 3885
>
> |

```

---

---

---

---

---

---

---

---

- Comando order

√ Fornece número de ordem de cada elemento do vetor

√ `order(consumo.pos)`

```

>
> order(consumo.pos)
[1] 3 1 2 4 5
>
>
> ordem<-order(consumo.pos)
>
>
> consumo.pos[ordem]
[1] 3885 3910 4220 5160 5645
>
>
> consumo.pre[ordem]
[1] 5640 5260 5470 6180 6390
>
> |

```

---

---

---

---

---

---

---

---

### Sintaxe – Alguns Pontos Importante

	Uso
( )	Para funções: como em $f(x)$ Priorização operações: como em $3*(4+2)$
[ ]	Para indexação: Como em <code>vetor[3]</code>

---

---

---

---

---

---

---

---

### Referências

---

---

---

---

---

---

---

---

### Bibliografia

- Dalgaard, P. (Spring)  
*Introductory Statistics with R*

---

---

---

---

---

---

---

---