

## Introdução ao R – Fundamentos

Lupércio França Bessegato  
Dep. de Estatística/UFJF

## Apresentação



- Lupércio F. Bessegato
  - √ Professor do Departamento de Estatística/UFJF
  - √ Membro permanente do corpo docente do Mestrado Acadêmico de Administração/UFJF
  - √ Linhas de pesquisa e extensão em Análise e Modelagem de Dados Multivariados
  - √ Site: [http://www.ufjf.br/lupercio\\_bessegato](http://www.ufjf.br/lupercio_bessegato)
  - √ E-mail: [lupercio.bessegato@ufjf.edu.br](mailto:lupercio.bessegato@ufjf.edu.br)

Introdução ao R - Fundamentos - 2018

2

## Roteiro Geral



1. Fundamentos da linguagem R
2. Visualização e descrição de dados
3. Inferência estatística básica
4. Referências

Introdução ao R - Fundamentos - 2018

3

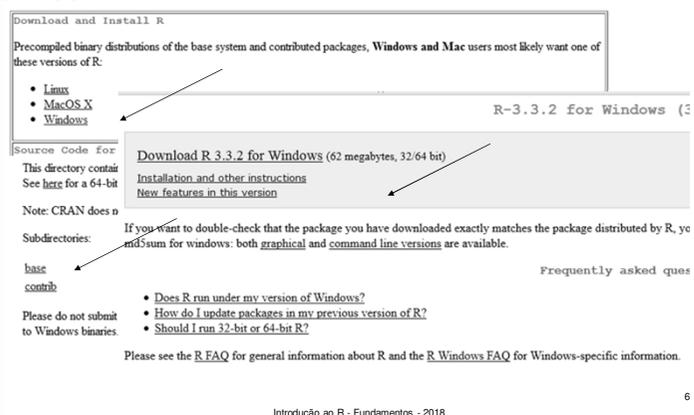
## Fundamentos da Linguagem R

## 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:  
 $\sqrt{\text{http://cran.r-project.org}}$

5

## Download do Programa



6

## 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
> |
    
```

7

- Todas as funções do R devem ser digitadas em letras **minúsculas**  
 $\sqrt{\text{O R é sensível a letras maiúsculas e minúsculas.}}$
- Todas as palavras-chaves do R estão em letras minúsculas
- R usa um ponto “.” em vez de virgula “,” quando há números com casas decimais.

8



## Help na Web



- R Project  
√ <http://www.r-project.org>
- Tutorial de Introdução ao R  
√ <http://ecologia.ib.usp.br/bie5782/doku.php?id=start>
- R Graphical Manual  
√ [http://www.imsbio.co.jp/RGM/R\\_image\\_list?page=1&sort=-](http://www.imsbio.co.jp/RGM/R_image_list?page=1&sort=-)

Introdução ao R - Fundamentos - 2018

13

## Armazenando dados



- Tipos de dados:
  - √ Numéricos
  - √ Caracteres: compostos por letras ou palavras.
  - √ Lógicos
- Quando os dados são armazenados, eles são chamados de objetos.
- Para armazenar um objeto basta utilizar o símbolo “<- ” ou “=”.

Introdução ao R - Fundamentos - 2018

14

## Tipos de Objetos



- Variável
- Vetor
- Matriz
- Data Frame
- Array
- Lista
- Fatores

Introdução ao R - Fundamentos - 2018

15

## Comando de Atribuição – Variável



- Atribuição de valor à variável  $x$  (ou qualquer outro ‘nome’)

```
x <- 4      # atribui o valor 4 à variável x
x1 <- -4    # atribui o valor -4 à variável x1
x2 <- 6     # atribui o valor 6 à variável x2
x; x1; x2   # print no console das variáveis x, x1 e x2
```

√ Evite usar acentos ou ç na denominação das variáveis

- Pode-se aplicar funções e operações à variável

```
z <- x1 + x2 # atribui à variável z a soma de x1 e x2
```

Introdução ao R - Fundamentos - 2018

16

## Operações com as Variáveis

- Calcular:  $x^{x1} + \frac{z}{x2}$

```
x^x1 + z/x2 # calcula a expressão
[1] 0.3372396 # resultado da expressão
```

Introdução ao R - Fundamentos - 2018 17

## Nomes Inválidos

```
24e <- 4 # nome do objeto começa com números
12 <- 2 # nome do objeto é um número
e*2 <- 6 # nome do objeto contém operador matemático
x -> 5 # atribuição no sentido incorreto
```

- Importante:
  - √ É possível atribuir no sentido inverso

```
6 -> y # atribuição com seta invertida
```

Introdução ao R - Fundamentos - 2018 18

## Apagar Variável

- Comando `rm( variavel )`
  - √ Cuidado! Não tem *undo*

```
rm(x, x1) # Apaga com um só comando
rm(x2); rm(z) # Apaga com vários comandos
```

- Apagar tudo!

```
rm(list = ls(all = TRUE)) # Apaga todos os objetos
```

Introdução ao R - Fundamentos - 2018 19

## Limpeza do Console

- Para limpar o console usa-se CTRL + L
- 

Introdução ao R - Fundamentos - 2018 20

## Vetores

- Coleção unidimensional de dados pontuais de mesmo tipo (números, strings, lógico)
- Comando para armazenar dados em vetor  
 $\checkmark$  `c(dados)` # concatenar

```

xNum <- c(1,2,5.3,6,-2,4) # vetor numérico
xCar <- c("um", "dois", "três") # vetor de caracteres
xLog <- c(TRUE, TRUE, TRUE, FALSE, TRUE, FALSE) # vetor lógico
    
```

Introdução ao R - Fundamentos - 2018 21

## Vetores – Indexação

- Elementos particulares de uma estrutura de dados

```

xNum[2] # retorna o 2º elemento do vetor
xCar[c(1,3)] # retorno o 1º e o 3º elementos do vetor
    
```

Introdução ao R - Fundamentos - 2018 22

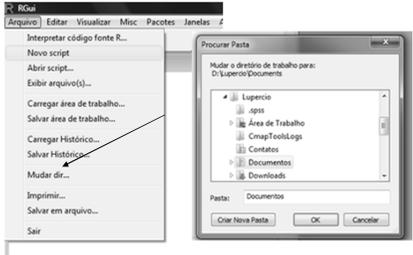
## Diretório de Trabalho

- Comando para verificar o diretório de trabalho que o R está usando:  
 $\checkmark$  `getwd()`
- Ideal sempre deixar scripts e dados de trabalho no mesmo diretório!
- Comando para mudar o R para seu diretório  
 $\checkmark$  `setwd("caminho_ate_diretorio")`  
 $\checkmark$  Ex.: "D:/Lupercio/Documents"

Introdução ao R - Fundamentos - 2018 23

## Mudança Diretório – Barra de Ferramentas

- Sugestão:
  - $\checkmark$  Sempre mude para seu diretório de trabalho quando iniciar a sessão em R
  - $\checkmark$  Guarde nele seus dados, gráficos, scripts, etc



Introdução ao R - Fundamentos - 2018 24

## Outros Comandos Úteis



- Diretório de trabalho e *workspace*

```
> getwd() # verifica caminho até diretório de trabalho
[1] "D:/Lupercio/Dropbox/!Current/!Docencia/Est032_Pacotes_R/R_Project/minicurso "

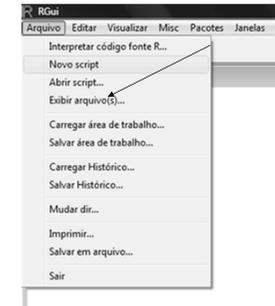
> dir() # verifica arquivos no diretório de trabalho
[1] "AirPassengers.csv" "analise-saida.txt" "dadosfic.csv"
[4] "exemplo02.txt" "gam01.txt" "letras.rdata"
[7] "mtcars.csv" "mtcars.sps" "mtcars.txt"
[10] "novo.R" "script.R" "script_introducao.R"
[13] "solo.csv" "test2.txt" "tp.txt"
[16] "turma.csv"

> ls() # verifica objetos no workspace
[1] "xCar" "xLog" "xNum"
```

## Uso de Script



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



## Compilação do Script



- 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

## Uso do Script



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

## Vetores Numéricos



- Comando concatenar: `c(dados)`
  - √ Criar o vetor `ano`
    - Valores: 2013, 2014, 2015, 2016

```
> ano <- c(2013, 2014, 2015, 2016)
> ano
[1] 2013 2014 2015 2016
```

- √ Criar o vetor `temp` (temperatura em °F)
  - Valores: 51,9; 51,8; 51,9; 53

```
> temp <- c(51.9, 51.8, 51.9, 53) # temperatura média em F
> temp
[1] 51.9 51.8 51.9 53.0
```

## Outras Funções para Criar Vetores



- Função `seq`
  - √ Lista sequência de números que quiser, no intervalo que quiser
  - √ `seq(from=1, to=1, length.out = NULL, by = ((to - from)/(length.out - 1)), along.with = NULL)`
    - `from`, `to`: início e fim da sequência
    - `by`: incremento da sequência
    - `length.out`: tamanho desejado da sequência
    - `along.with`: comprimento do objeto do argumento

## Criação de vetores



- Criando sequências

```
> seq(1, 100, 1) # sequência de 1 a 100, com intervalo de 1 unidade
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
[19] 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
[37] 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
[55] 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
[73] 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
[91] 91 92 93 94 95 96 97 98 99 100

> seq(1, 100, 2) # sequência de 1 a 100, com intervalo de 2 unidades
[1] 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49
[26] 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99

> seq(1, 100, 10) # sequência de 1 a 100, com intervalo de 10 unidades
[1] 1 11 21 31 41 51 61 71 81 91
```

## Exemplos



- Mais exemplos

```
> seq(20, 1, -1) # sequência decrescente de 20 até 1
[1] 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

> seq(1, 20) # sequência de 1 a 20, com intervalo de 1 unidade
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

> 1:20 # sequência de 1 a 20, com intervalo de 1 unidade
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

- Sequências podem ser armazenadas

```
> sequencia <- 1:20 # armazenando sequência em um vetor
> sequencia
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

## Outras Funções para Criar Vetores

- Função `rep`
  - √ Lista números repetidos, quantos números quiser com quantas repetições quiser
  - √ `rep(x, times = 1, length.out = NA, each = 1)`
    - `x`: um vetor ou um fator ou uma lista
    - `times`: vetor com o nº de vezes de repetições de cada elemento
    - `length.out`: tamanho desejado da sequência
    - `each`: cada elemento de `x` é repetido `each` vezes
    - `along.with`: comprimento do objeto do argumento

33

Introdução ao R - Fundamentos - 2018

## Repetindo Sequências

- Criando repetições:

```

> rep(1, 10) # número 1 é repetido 10 vezes
[1] 1 1 1 1 1 1 1 1 1 1
> rep(1:5, 2) # sequência de 1 a 5 é repetida 2 vezes
[1] 1 2 3 4 5 1 2 3 4 5
> rep(c(1, 4), 3) # números 1 e 4 repetidos 3 vezes, alternadamente
[1] 1 4 1 4 1 4
> rep(c(1, 4), each = 3) # números 1 e 4 repetidos 3 vezes, sequencialmente
[1] 1 1 1 4 4 4
    
```

√ O comando `rep` também funciona com caracteres e valores lógicos

34

Introdução ao R - Fundamentos - 2018

## Alguns Vetores Internos

```

> LETTERS
[1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S"
[20] "T" "U" "V" "W" "X" "Y" "Z"

> letters
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
[20] "t" "u" "v" "w" "x" "y" "z"

> month.name
[1] "January" "February" "March" "April" "May" "June"
[7] "July" "August" "September" "October" "November" "December"

> month.abb
[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov" "Dec"
    
```

35

Introdução ao R - Fundamentos - 2018

## Algumas Estatísticas Descritivas

- Média:
  - √ `mean(dados)`
- Desvio-padrão:
  - √ `sd(dados)`
- Variância
  - √ `var(dados)`
- Mediana
  - √ `median(dados)`

36

Introdução ao R - Fundamentos - 2018

- Estatísticas descritivas do vetor *temp*

```
# Estatísticas descritivas do vetor temp

mean(temp) # media do vetor temp
sd(temp)   # desvio padrão do vetor temp
var(temp)  # variância do vetor temp
sd(temp)^2
median(temp) # mediana do vetor temp
sum(temp)   # soma dos elementos do vetor temp
length(temp) # quantidade de elementos do vetor temp
```

√ E o coeficiente de variação?

## Aritmética de Vetores

- R manipula vetores como simples objetos
- Cálculo da temperatura em °C

$$^{\circ}C = \frac{5(^{\circ}F - 32)}{9}$$

```
> celsius <- (5/9) * (temp - 32)
> celsius
[1] 11.05556 11.00000 11.05556 11.66667
```

- Valores arredondados

```
> round(celsius, 2)
[1] 11.06 11.00 11.06 11.67
```

- Pode-se digitar sequência de comandos em um editor de textos e depois colar no console para execução das tarefas

```
sum(temp); length(temp); xbarra<-sum(temp)/length(temp)
```

```
> sum(temp)
[1] 208.6

> length(temp)
[1] 4

> xbarra <- sum(temp)/length(temp)
> xbarra
[1] 52.15
```

## Vetores Lógicos

- Podem tomar os valores TRUE ou FALSE (ou NA)

√ c(T,T,F,T)

- Expressões de Relação

```
> temp > 52
[1] FALSE FALSE FALSE TRUE

> temp[temp > 52]
[1] 53

> ano[temp > 52]
[1] 2016
```

## 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)

41

Introdução ao R - Fundamentos - 2018

## Matrizes

- Todas as colunas tem de ter o mesmo comprimento e conter o mesmo tipo de dados (numérico, caracter, etc.)
- `matrix(vector, nrow=r, ncol=c, byrow=F, dimnames=list(char_vector_rownames, char_vector_colnames))`
  - √ `byrow=TRUE`: matriz deve ser montada pelas linhas.
  - √ `dimnames`: nomes para linhas e colunas (opcional)

42

Introdução ao R - Fundamentos - 2018

## Exemplos

- Criação de matrizes

```
# gera matriz numérica 5 x 4
mat1 <- matrix(1:20, nrow = 5, ncol = 4)
mat1

# outro exemplo
celulas <- c(1, 26, 24, 68)
rnomes <- c("R1", "R2")
cnomes <- c("C1", "C2")
my.mat <- matrix(celulas, nrow = 2, ncol = 2, byrow = TRUE,
  dimnames =list(rnomes, cnomes))
```

43

Introdução ao R - Fundamentos - 2018

## Exemplos

- Matrizes a partir de combinação de objetos

```
x <- c(11, 12, 13)      # cria um vetor `x` com 3 valores
y <- c(55, 33, 12)     # cria um vetor `y` com 3 valores
rbind(x, y)            # combina os vetores em linhas (2 x 3)
cbind(x, y)            # combina os vetores em colunas (3 x 2)
dados <- cbind(ano, temp, celsius) # cria matriz de dados
dados
```

44

Introdução ao R - Fundamentos - 2018

• Atribuindo nomes às linhas e colunas



```
> mat2 <- matrix(1:12, nrow = 3, byrow = F)
> rownames(mat2) <- c("Primeira", "Segunda", "Terceira")
> colnames(mat2) <- LETTERS[1:4]
> mat2
      A B C D
Primeira 1 4 7 10
Segunda  2 5 8 11
Terceira 3 6 9 12
```

**Desconstrução de Matriz**



• Podemos desconstruir uma matriz aplicando a função `c`

√ Combina todas as colunas em um vetor

```
c(my.mat) # Combina todas as colunas em um vetor
```

**Matrizes – Indexação**



• Identificação de linhas, colunas ou elementos usando subscritos.

```
# Cria matriz com 20 números aleatórios de normal padrão
x <- matrix(rnorm(20), ncol = 4)
x
x[,4] # 4ª coluna da matriz
x[3,] # 3ª linha da matriz
x[2:4,1:3] # linhas 2,3 e 4 das colunas 1,2 e 3
x[c(1,3), c(2, 4)] # linhas 1 e 3 das colunas 2 e 4
```

**Arrays**



• Arrays são similares a matrizes, mas podem ter mais de duas dimensões.

## Fatores



- Estrutura de dados para variável categórica:
  - √ Há análises que o R precisar distinguir códigos categóricos dos numéricos
  - √ Ex.: média de variável categórica

```
# variável sexo com 20 componentes "masc" e
# 30 "fem"
sexo <- c(rep("masc", 20), rep("fem", 30))
fsexo <- factor(sexo)
# armazena sexo como 20 1's and 30 2's e associa
# internamente 1 = fem, 2 = masc (alfabeticamente)
# R agora trata sexo com uma variável nominal
summary(fsexo)
```

Introdução ao R - Fundamentos - 2018

49

## Variável Categórica Codificada Numericamente



```
# vetor codificando nível de dor de 5 pacientes
dor <- c(0, 3, 2, 2, 1)
# transforma vetor em fator (categórica)
fdor <- factor(dor, level = 0:3)

# muda nome dos níveis do fator
levels(fdor) <- c("nenhum", "leve", "medio", "grave")
fdor
# extrai os nomes dos níveis do fator
levels(fdor)
# extrai códigos numéricos dos níveis do fator
as.numeric(fdor)
```

Introdução ao R - Fundamentos - 2018

50

## Fator Ordenado



- Usado para representação de variável ordinal

```
# Cria string denominada nivel (nível de negócios).
nivel <- c("baixo", "medio", "baixo", "baixo", "baixo", "baixo",
"medio", "baixo", "medio", "medio", "medio", "medio", "medio",
"alto", "alto", "baixo", "medio", "medio", "baixo", "alto")

is.factor(nivel)           # verifica se nivel é fator
is.character(nivel)       # verifica se nivel é string
fnivel <- factor(nivel)    # transforma nivel em fator
is.factor(fnivel)         # verifica se nivel é fator
levels(fnivel)            # extrai os nomes dos níveis do fator

# Cria fator com a ordem correta dos níveis
fnivel.ord <- factor(nivel, levels = c("baixo", "medio", "alto"))
levels(fnivel.ord)
# ordena fator
fnivel.ord2 <- ordered(fnivel, levels = c("baixo", "medio", "alto"))
levels(fnivel.ord2)
```

Introdução ao R - Fundamentos - 2018

51

## Listas



- Coleção ordenada de objetos (componentes)
  - √ Permite que reunir objetos de diferentes tipos sob o mesmo nome.

```
# Lista com 4 componentes
# uma string, um vetor numérico, uma matriz e um escalar
lista <- list(nome = "Thiago", numerico = temp, matriz = dados, idade = 62)
lista
lista$matriz

# Criação de lista contendo duas listas
listao <- c(lista1, lista2)
```

Introdução ao R - Fundamentos - 2018

52

## Listas – Indexação



- Identificação dos objetos de uma lista usando `[[ ]]`.

```
minhalista[[2]]           # 2º componente da lista
minhalista[["nome_dele"]] # componente denominado nome_dele na lista
```

```
> lista[[2]]
[1] 51.9 51.8 51.9 53.0

> lista["matriz"]
$matriz
  ano temp celsius
[1,] 2013 51.9 11.05556
[2,] 2014 51.8 11.00000
[3,] 2015 51.9 11.05556
[4,] 2016 53.0 11.66667
```

Introdução ao R - Fundamentos - 2018

53

## Data Frames



- É mais geral que uma matriz
  - √ Colunas diferentes de mesmo comprimento podem ter diferentes tipos de dados (numérico, caracter, lógico, fator, etc.)

```
d <- c(1,2,3,4)
e <- c("vermelho", "branco", "vermelho", NA)
f <- c(TRUE, TRUE, TRUE, FALSE)
meu.banco <- data.frame(d,e,f)
# nome das variáveis
names(meu.banco) <- c("ID", "Cor", "verificou")
```

Introdução ao R - Fundamentos - 2018

54

## Data Frames – Indexação



- Há várias maneiras para identificar os elementos de um data frame

```
meu.banco[2:3]           # colunas 2 e3 do data frame
meu.banco[c("ID", "verificou")] # columns ID e do data frame
meu.banco$Cor           # variable verificou do data frame
```

Introdução ao R - Fundamentos - 2018

55

## Exemplo – Duelos ‘Esportivos’



- Alturas dos concorrentes

```
> vence <- c(185, 182, 182, 188, 188, 188, 185, 185, 177, 182, 182, 193,
+ 183, 179, 179, 175)
> perde <- c(175, 193, 185, 187, 188, 173, 180, 177, 183, 185, 180, 180,
+ 182, 178, 178, 173)
```

- √ Quantidade de duelos

```
> length(vence)
[1] 16
```

- √ Anos dos duelos

```
ano <- seq(from = 2008, to = 1948, by = -4)
```

Introdução ao R - Fundamentos - 2018

56

## Indexação

- Altura vencedor #5

```
> vence[5]
[1] 188
```

- Alturas vencedores #1, 3, 5

```
> vence[c(1, 3, 5)]
[1] 185 182 188
```

- Alturas 3 primeiros vencedores

```
> vence[1:3]
[1] 185 182 182
```

57

- Todos vencedores, exceto # 1, 3 e 5

```
> vence[-c(1, 3, 5)]
[1] 182 188 188 185 185 177 182 182 193 183 179 179 175
```

√ Alternativa:

```
> criterio <- c(1, 3, 5)
> vence[-criterio]
[1] 182 188 188 185 185 177 182 182 193 183 179 179 175
```

58

## √ Médias

```
> mean(vence)
[1] 183.3125

> mean(perde)
[1] 183.3125
```

- √ Perdedores são mais baixos?
- √ Diferenças entre vencedor e derrotado

```
> diferenca <- vence - perde
> diferenca
[1] 10 -11 -3 1 0 15 5 8 -6 -3 2 13 1 1 1 2
```

59

- Criação de data frame

```
> dados.duelo <- data.frame(ano, vence, perde, diferenca)
> dados.duelo
  ano vence perde diferenca
1 2008  185  175      10
2 2004  182  193     -11
3 2000  182  185      -3
4 1996  188  187       1
5 1992  188  188       0
6 1988  188  173      15
7 1984  185  180       5
8 1980  185  177       8
9 1976  177  183      -6
10 1972  182  185      -3
11 1968  182  180       2
12 1964  193  180      13
13 1960  183  182       1
14 1956  179  178       1
15 1952  179  178       1
16 1948  175  173       2
```

- Variáveis acessíveis usando notação \$

```
> dados.duelo$diferenca
[1] 10 -11 -3 1 0 15 5 8 -6 -3 2 13 1 1 1 2
```

60



- Indexação pelo do Data frame
- √ Altura do vencedor #5

```
> dados.duelo$vence[5]
[1] 188
```

- √ Todas as informações do duelo #5:

```
> dados.duelo[5, ]
  ano vence perde diferenca
5 1992  188  188      0
```

- √ Alturas dos 3 primeiros vencedores

```
> dados.duelo[1:3, 2]
[1] 185 182 182
```

Introdução ao R - Fundamentos - 2018 61



## Seleção Condicional

- Anos em que o vencedor é mais alto

```
> alto.vence <- vence > perde
> alto.vence
[1] TRUE FALSE FALSE TRUE FALSE TRUE TRUE TRUE FALSE FALSE TRUE TRUE
[13] TRUE TRUE TRUE TRUE

> ano[alto.vence]
[1] 2008 1996 1988 1984 1980 1968 1964 1960 1956 1952 1948
```

- Anos para altura de vencedor entre 182,5 e 190

```
> intervalo <- vence > 182.5 & vence < 190
> ano[intervalo]
[1] 2008 1996 1992 1988 1984 1980 1960
```

Introdução ao R - Fundamentos - 2018 62



## Ordenação

- Comando sort: ordena os elementos do vetor
- √ Default é crescente

```
> vence
[1] 185 182 182 188 188 188 188 185 185 177 182 182 193 183 179 179 175
>
> sort(vence)
[1] 175 177 179 179 182 182 182 182 183 185 185 185 188 188 188 193
```

- Ordenação decrescente

```
> sort(vence, decreasing = TRUE)
[1] 193 188 188 188 185 185 185 183 182 182 182 182 179 179 177 175
```

Introdução ao R - Fundamentos - 2018 63



- Comando order
- √ Fornece a ordem de cada elemento do vetor

```
> order(vence)
[1] 16 9 14 15 2 3 10 11 13 1 7 8 4 5 6 12
>
> ordem <- order(vence)
> vence[ordem]
[1] 175 177 179 179 182 182 182 182 183 185 185 185 188 188 188 193
```

Introdução ao R - Fundamentos - 2018 64

- Tabelas de frequências

√ Alturas de vencedores maiores que de derrotados

```
> table(alto.vence)
alto.vence
FALSE TRUE
5 11

> table(alto.vence)/length(vence) * 100
alto.vence
FALSE TRUE
31.25 68.75
```

√ Alturas de vencedores entre 182,5 cm e 190 cm

```
> table(intervalo)
intervalo
FALSE TRUE
9 7

> table(intervalo)/length(vence) * 100
intervalo
FALSE TRUE
56.25 43.75
```

Introdução ao R - Fundamentos - 2018 65

- Visualização gráfica das diferenças alturas

```
> barplot(rev(diferenca), xlab = "Anos de disputa - 1948 a 2008",
+ ylab = "Diferença na altura, em cm")
```

√ Diferenças ao longo do tempo

Introdução ao R - Fundamentos - 2018 66

- Visualização gráfica das diferenças alturas

```
> plot(vence, perde, xlab = "Altura do vencedor, em cm",
+ ylab = "Altura do perdedor, em cm")
> abline(0, 1, lty = 2)
> text(vence, perde, ano, cex = 0.7, pos = 4)
```

√ Labels fora do gráfico  
√ Há empates

Introdução ao R - Fundamentos - 2018 67

```
> # ajustando escala do eixo x
> plot(vence, perde, xlim = c(175, 195), xlab = "Altura do vencedor, em cm",
+ ylab = "Altura do perdedor, em cm")
> abline(0, 1, lty = 2)
> text(vence, perde, ano, cex = 0.7, pos = 4)
```

√ Ajustado eixo x

Introdução ao R - Fundamentos - 2018 68

## Funções Úteis



- Utilizem os comandos em objetos criados

```
length(objeto) # numero de elementos ou componentes
str(objeto)    # estrutura de um objeto
class(objeto)  # classe ou tipo de um objeto
names(objeto)  # nomes

c(objeto,objeto,...) # combina objetos em um vetor
cbind(objeto, objeto, ...) # combina objetos como colunas
rbind(objeto, objeto, ...) # combina objetos como linhas

objeto        # prints the object

ls()          # lista objetos em uso
rm(objeto)    # delete um objeto

novoobjeto <- edit(objeto) # edit copy and save as newobject
fix(objeto)   # edit in place
```

Introdução ao R - Fundamentos - 2018

69

## Criação de Data-frame



- Maneira diferente de criar data-frame
  - √ Comando `edit(data.frame())`
  - √ `dados.novo <- edit(data.frame())`
- É aberta janela com planilha e pode-se digitar informações do banco de dados.
- Para nomear a 1ª. variável, clica-se em “var1”
  - √ Janela para nomear a variável e seu tipo (numérica ou caractere).

Introdução ao R - Fundamentos - 2018

70

## Exemplo



```
edit(data.frame())
```

**R Editor de dados**

Arquivo Editar Ajuda

	var1	var2	var3	var4	var5
1					
2					
3					

Introdução ao R - Fundamentos - 2018

71

## Carregando Pacotes



- Pacotes:
  - √ Conjuntos de funções específicas do R
  - √ No repositório do R está armazenada uma quantidade muito grande de pacotes que tem funções para um certo conjunto de tarefas
  - √ Para usar um pacote:
    - baixar o pacote (download) do repositório
    - carregar o pacote na sua área de trabalho.

Introdução ao R - Fundamentos - 2018

72

## Pacotes

- Quais pacotes estão disponíveis na sua instalação de R?
  - √ `library()`
- Interface hipertexto de ajuda:
  - √ `help.start()`
  - √ Escolher o link “Packages”
  - √ Clique no nome de um dos pacotes
    - Lista todos os objetos que este pacote contém.

73

## • Página do R Project

Package Index

---

Packages in C:\Program Files\R\R-3.3.1\library

A B C D E E F G H I J K L M N O P Q R S T U V W X Y Z

<p><a href="#">abind</a></p> <p><a href="#">abundant</a></p> <p><a href="#">acepack</a></p> <p><a href="#">ade1</a></p> <p><a href="#">asbio</a></p> <p><a href="#">assertthat</a></p> <p><a href="#">base</a></p> <p><a href="#">base64enc</a></p> <p><a href="#">BB</a></p> <p><a href="#">BH</a></p> <p><a href="#">BHH2</a></p> <p><a href="#">BiasedUm</a></p> <p><a href="#">BioGenerics</a></p> <p><a href="#">BioInstaller</a></p>	<p>Combine Multidimensional Arrays</p> <p>Abundant regression and high-dimensional principal fitted components</p> <p>ace() and avas() for selecting regression transformations</p> <p>Analysis of Ecological Data : Exploratory and Euclidean Methods in Environmental Sciences</p> <p>A Collection of Statistical Tools for Biologists</p> <p>Easy pre and post assertions.</p> <p>The R Base Package</p> <p>Tools for base64 encoding</p> <p>Solving and Optimizing Large-Scale Nonlinear Systems</p> <p>Boost C++ Header Files</p> <p>Useful Functions for Box, Hunter and Hunter II</p> <p>Biased Um Model Distributions</p> <p>S4 generic functions for Bioconductor</p> <p>Install/Update Bioconductor, CRAN, and github Packages</p>
--	--

74

- Quais pacotes estão carregados na sua sessão?
  - √ `search()`
- Instalação de pacote direto do R
  - √ `install.packages("vegan")`

75

## Exemplo

- Geração de amostra aleatória:
 

```
# 15 números aleatórias de uma distribuição normal, com média 1 e
# desvio-padrão=3

x1 <- rnorm(n = 15, mean = 1, sd = 3)
hist(x1) # histograma de frequência
Hist(x1, freq = F) # histograma de densidade
truehist(x1) # outro tipo de histograma
```
- Uso de função do pacote MASS:
 

```
search() # pacotes disponíveis área de trabalho
library(MASS) # carrega pacote MASS
truehist(x1)
help(package = MASS) # ajuda sobre o pacote
```

76

## Demos

- Oferece demonstrações de como usar funções

```
# todos os demos (pacotes carregados)
demo()
# todos demos (pacotes instalados)
demo(package = .packages(all.available = TRUE))
demo(plotmath)      # demo sobre anotações em gráfico
demo(graphics)      # demos sobre tipos de gráficos
```

Introdução ao R - Fundamentos - 2018 77

## Help

- Bom site para verificar outros tipos de ajuda  
<http://stackoverflow.com/questions/15289995/how-to-get-help-in-r>

Introdução ao R - Fundamentos - 2018 78

## Conjuntos de Dados do R

- R traz vários conjuntos de dados internos, que são geralmente usados em demos ou exemplos
- Comando para ver a lista dos conjuntos de dados carregados:  
√ `data()`

Introdução ao R - Fundamentos - 2018 79

## Carregamento Built-in Data Set

- Carregamento do conjunto de dados:  
√ `mtcars`: Motor Trend Car Road Tests

```
# Conjunto de dados: mtcars: Motor Trend Car Road Tests
data(mtcars)      # carregamento
head(mtcars, 8)   # Print das primeiras 8 linhas
help(mtcars)      # informações sobre o banco
```

- Manipulação do conjunto de dados

```
nrow(mtcars)      # Número de linhas (observações)
ncol(mtcars)      # Número de colunas (linhas)
str(mtcars)        # estrutura do objeto
```

Introdução ao R - Fundamentos - 2018 80

• Colunas como vetores

```
mtcars$mpg           # acesso à variável mpg
mtcars$displ         # acesso à variável displ
pairs(mtcars, main = "Conjunto mtcars")
plot(mpg, displ)
# acesso às variáveis sem especificar banco
attach(mtcars)
plot(mpg, displ)     # plot das variáveis mpg e displ
objeto <- lm(displ ~ mpg) # regressão linear
objeto               # resultados
summary(objeto)     # ajuste do modelo de regressão
abline(lm(displ ~ mpg)) # plot da reta de regressão
# estratificação do plot pela variável cyl
coplot(mpg ~ displ | as.factor(cyl), data = mtcars,
       panel = panel.smooth, rows = 1)
detach(mtcars) # interromper acesso (final do trabalho)
```

Exportação de Bancos de Dados

• Dados exportados para outras extensões

```
# para arquivo texto (delimitado por Tab)
# formato do R
write.table(mtcars, "mtcars.txt", sep="\t")
# muda ponto decimal
write.table(mtcars, "mtcars.txt", dec = ",", sep="\t")

# para arquivo csv (Comma Separated Value)
write.csv(mtcars, "mtcars.csv") # formato do R
write.csv2(mtcars, "mtcars.csv") # formato brasileiro

# para planilha Excel
library(xlsx) # requer versão recente do R
write.xlsx(mtcars, "mtcars.xlsx")
```

Exportação de Bancos de Dados

• Dados exportados para outras extensões

```
# para arquivo do SPSS
# escreva arquivo de texto e um programa SPSS para sua leitura
library(foreign)
write.foreign(mtcars, "mtcars.txt", "mtcars.sps", package="SPSS")

# Para arquivo SAS
# escreva arquivo de texto e um programa SAS para sua leitura
library(foreign)
write.foreign(mtcars, "mtcars.txt", "mtcars.sas", package="SAS")

# Para arquivo Stata
# exportação de data frame para formato binário do Stata
library(foreign)
write.dta(mtcars, "mtcars.dta")
```

Comando sink

• Desvia as saídas do R para uma conexão (e desliga a conexão).

```
# Grava saída de uma análise
sink('analise-saida.txt')
set.seed(12345) # semente aleatoria
x <- rnorm(10, 10, 1)
y <- rnorm(10, 11, 1)
# alguma saída
cat(sprintf("x tem %d elementos:\n", length(x)))
print(x)
cat("y =", y, "\n")
cat("=====\n")
cat("Teste-t entre x e y\n")
cat("=====\n")
t.test(x,y)
# para o arquivo
sink()
```

## Importação de Dados

- Importação para o R de dados disponíveis em formato eletrônico
- Comandos
  - √ read.table
  - √ read.csv
  - √ read.csv2
  - √ outros

Introdução ao R - Fundamentos - 2018

85

## Comando read.table

- Importação de dados em formato texto (arquivo do tipo ASCII)

```
# arquivo sem cabeçalho
ex01 <- read.table("gam01.txt")
head(ex01)
# arquivo com cabeçalho na 1a. linha
ex02 <- read.table("exemplo02.txt", head=T)
head(ex02)
# arquivo com campos separados por : e decimais, por vírgula
ex03 <- read.table("dadosfic.csv", head=T, sep=":", dec=",")
head(ex03)
# leitura direta pela web
ex04 <- read.table("http://www.leg.ufpr.br/~paulojus/dados/gam01.txt")
head(ex04)
# leitura de arquivo com informações em suas 1as. linhas
teste.data <- read.table("test2.txt", skip=4, header=TRUE, sep="\t")
head(teste.data)
```

Introdução ao R - Fundamentos - 2018

86

## Comando read.csv

- Importação de dados em formato csv (*Comma Separated Value*)

```
# arquivo csv formato inglês
aereas.data <- read.csv("AirPassengers.csv", header=TRUE)
head(aereas.data)
# arquivo csv - comando read.table
aereas.data2 <- read.table("AirPassengers.csv", header=TRUE, sep=",")
head(aereas.data2)
# arquivo csv - gravado em formato brasileiro
solo <- read.csv("solo.csv", header = TRUE, dec = ",", sep = ";")
head(solo)
# comando read.csv2 (leitura direta de csv em formato brasileiro)
solo2 <- read.csv2("solo.csv", header = TRUE)
head(solo2)
```

Introdução ao R - Fundamentos - 2018

87

## Importação do Excel

- Importação direta de planilhas com extensão xlsx

```
# lê a primeira guia da planilha meuexcel.xlsx
# nomes das variáveis na primeira linha
library(xlsx)
dados <- read.xlsx("meuexcel.xlsx", 1)

# lê a guia na planilha chamada minhaguia
dados <- read.xlsx("meuexcel.xlsx", sheetName = "minhaguia")
```

Introdução ao R - Fundamentos - 2018

88

## Importação do SPSS



- Importação direta de arquivos com extensão sav

```
# exportar os dados no SPSS
get file='meusdados.sav'.
export outfile='meusdados.por'.

# no R
library(Hmisc)
dados <- spss.get("meusdados.por", use.value.labels=TRUE)
# última opção converte níveis em fatores no R
```

Introdução ao R - Fundamentos - 2018

89

## Importação do Stata



- Importação direta de arquivos com extensão dta

```
# importação do arquivo do Stata
library(foreign)
dados <- read.dta("meusdados.dta")
```

Introdução ao R - Fundamentos - 2018

90

## Importação do SAS



- Importação direta de arquivos com extensão xpt

```
# exportar os dados no SAS
libname out xport 'c:/mydata.xpt';
data out.mydata;
set sasuser.mydata;
run;

# in R
library(Hmisc)
dados <- sasxport.get("c:/mydata.xpt")
# variáveis de caracteres são convertidas para fatores no R
```

Introdução ao R - Fundamentos - 2018

91

## Referências

### Bibliografia Recomendada



- ALBERT, J.; RIZZO, M. *R by Example*. Springer, 2012.
- CHAPMAN, C.; FEIT, E. M. *R for marketing research and analytics*. Springer, 2015.
- KLEIBER, C.; ZEILEIS, A. *Applied econometrics with R*. Springer, 2008.
- DALGAARD, P. *Introductory statistics with R*. Springer, 2008.