

O R

Frederico R. B. Cruz

Departamento de Estatística
Universidade Federal de Minas Gerais
E-mail: fcruz@est.ufmg.br

Junho de 2022

Introdução

- disponibiliza um ambiente flexível e poderoso para implementação de novas técnicas estatísticas;
- suíte para análise de dados e apresentação gráfica;
- disponibiliza:
 - extensiva coleção de ferramentas;
 - modelos lineares e não lineares;
 - análise gráfica;
 - uma linguagem de programação orientada por objetos, facilmente extensível;
- o R é um “ambiente” \neq de “pacote”;
- o R pode ser obtido diretamente no seu site:
<http://www.r-project.org/>

Aspectos Gerais I

- visão geral

- linha de comandos

```
> 2 + 3 
```

```
[1] 5
```

```
> sqrt(3/4)/(1/3 - 2/pi^2) 
```

```
[1] 6.6265
```

```
> dados <- c(95,90,85) 
```

```
> dados 
```

```
[1] 95 90 85
```

```
> mean(dados) 
```

```
[1] 90
```

```
> m <- mean(dados); v <- var(dados) 
```

```
> m/sqrt(v) 
```

```
[1] 18
```

Aspectos Gerais II

- criação de funções

```
std.dev <- function(x) {  
  sqrt(var(x))  
}
```

- tipos de dados disponíveis

- básicos

logical → binário (T ou F)

numeric → reais

complex → $a + bi$

character → caracteres

- compostos

vector → vetor unidimensional de mesmo tipo básico

matrix → matriz bidimensional de mesmo tipo básico

list → “vetor” unidimensional de tipos básicos diferentes

data frame → “matriz” bidimensional de tipos básicos diferentes

Aspectos Gerais III

- vetores

```
> dados <- c(95,90,85)
> dados
[1] 95 90 85
> dados[2]
[1] 90
> nomes <- c("A", "B", "C")
> nomes
[1] "A" "B" "C"
```

Aspectos Gerais IV

- matrizes

- exemplos

```
>notas.mat <- matrix(c(95,90,85,91,90,89),  
                    ncol=3, byrow=T)
```

```
>notas.mat
```

```
      [,1] [,2] [,3]  
[1,]   95   90   85  
[2,]   91   90   89
```

- funções para manipulação de matrizes

cbind → combina vetores como colunas de matrizes

rbind → combina vetores como linhas de matrizes

Aspectos Gerais V

- listas
 - elementos não precisam ser do mesmo tipo

```
>dados.list <- c("Aluno A", 95)
```

```
>dados.list
```

```
[1] "Aluno A" 95
```

- acesso aos componentes da lista

```
>dados.list[[2]]
```

```
[1] 95
```

- *data frames*

- linhas representam indivíduos
- colunas, representam variáveis
- funções para criação:

```
read.table("arq.ext", header=T)  
data.frame
```

Manipulação de Dados I

- acesso aos elementos

```
>dados.mat[3,2]
```

```
[1] 8
```

8 é o elemento da linha 3 e coluna 2

```
>dados.mat[3,]
```

```
[1] 3 8
```

```
>dados.mat[,2]
```

```
[1] 1 1 8
```

```
>dados.mat
```

```
      [,1] [,2]
```

```
[1,]    7    1
```

```
[2,]    5    1
```

```
[3,]    3    8
```


Manipulação de Dados II

- operações aritméticas

```
> a <- c(1,2,3,4)
```

```
> b <- c(2,3,4,5)
```

```
> a + b
```

```
[1] 3 5 7 9
```

```
> a * b
```

```
[1] 2 6 12 20
```

```
> bt <- t(b)
```

```
> a %*% b
```

```
> 40
```

As Funções do R I

- funções matemáticas

abs, max, min

cos, acos, sin, asin, tan, atan,

cosh, acosh, sinh, asinh, tanh, atanh,

ceiling(x), floor(x), trunc(x),

exp, log, log10, gamma, lgamma, sqrt

- exemplo:

```
> vector <- c(24, 25, 30, 28)
```

```
> min.elem <- min(vector)
```

```
> min.elem
```

```
[1] 24
```

As Funções do R II

- funções estatísticas
all, any, length, max, min,
mean, median, mode, prod, quantile,
sum, var, cor
- exemplo:
somar elementos de um vetor

```
> soma <- sum(vector)
```

```
[1] 107
```
- distribuições de probabilidade

tipo da operação	distribuição	
letra	nome	argumentos
r → gerar	norm	mean, sd
p → f.d.a.	unif	min, max
q → f.d.a. inversa	t	df
d → densidade	pois	lambda

As Funções do R III

- amostragem
 > `sample(vetor, tamanho, replace, prob)`
- funções úteis na análise exploratória de dados
 > `apply(matriz, MARGIN, FUNC(...))`
- exemplo:
 calcular a média de cada coluna da matriz abaixo:

```
> matriz
      [,1] [,2] [,3]
[1,]  10  20  28
[2,]   7  24  25
[3,]   3  18  17
[4,]   7  25  20
[5,]   7  NA  22
```

As Funções do R IV

```
> mean.mat <- apply(matriz, 2, mean)
```

aplica a cada coluna

```
> var.mat <- apply(matriz, 2, var)
```

```
> desv.pad.mat <- sqrt(var.mat)
```

- funções gráficas

```
> dev.new() ← abre janela gráfica
```

```
> par(mfrow=c(linhas, colunas))
```

```
> hist(x, probability=F, ...)
```

```
> boxplot (x1,x2,...,xn)
```

```
> ts.plot(ts(x1),ts(x2),...,ts(xn)) ← séries temporais
```

```
> plot(x,y) ← gráfico de dispersão
```

```
> pairs(x,y,z)
```

```
> abline(...) ← adiciona linha ao gráfico
```

```
> points(x,y) ← adiciona pontos
```

As Funções do R V

- > `lines(x,y)` ← adiciona segmentos
- > `graphics.off()` ← fecha todas as janelas gráficas

- funções definidas pelo usuário

```
nome <- function(argumentos) {  
  corpo  
}
```

- exemplo

```
> media <- function(x) {  
  return(mean(x))  
}
```

Programação em R I

- estrutura condicional

```
> conceito <- function(nota) {  
  if (nota < 40 ) {  
    conc <- "F"  
  } else if ( nota < 60 ) {  
    conc <- "E"  
  } else if ( nota < 70 ) {  
    conc <- "D"  
  } else if ( nota < 80 ) {  
    conc <- "C"  
  } ...  
  return(conc)  
}
```

Programação em R II

- estrutura de repetição

```
> media <- function(x) {  
  elementos <- length(x)  
  soma <- 0  
  for (cont in 1:elementos) {  
    soma <- soma + x[cont]  
  }  
  media <- soma/elementos  
  return(media)  
}
```

- estrutura de repetição II

```
while (condicao) {  
  tarefas  
}
```


Programação em R III

- estrutura de repetição III

```
repeat {  
  tarefas  
  if (condicao) break  
}
```

Um Exemplo I

- Problema (Venables & Ripley, 1994):
Criar uma função no R/S-PLUS para encontrar o estimador de máxima verossimilhança do parâmetro λ da distribuição de Poisson truncada em zero:

$$P(Y = y) = \frac{e^{-\lambda} \lambda^y}{(1 - e^{-\lambda}) y!}$$

que corresponde a observar apenas valores não nulos da distribuição de Poisson.

Um Exemplo II

- Solução:

A estimativa de máxima verossimilhança $\hat{\lambda}$ é encontrada igualando-se a média amostral à sua esperança:

$$\bar{y} = E(Y) = \frac{\hat{\lambda}}{1 - e^{-\hat{\lambda}}}.$$

O seguinte esquema iterativo pode ser utilizado:

$$\hat{\lambda}_{i+1} \leftarrow \bar{y}(1 - e^{-\hat{\lambda}_i}).$$

Um Exemplo III

```
# definindo a funcao
ajPoisson <- function(dados) {
  Eps <- 1E-03
  iMax <- 100
  yBarra <- mean(dados)
  i <- 0
  lambda.i.1 <- yBarra
  repeat {
    i <- i + 1
    lambda.i <- lambda.i.1
    lambda.i.1 <- yBarra*(1-exp(-lambda.i))
    cat("Iter= ", i, " Lambda= ", lambda.i.1, "\n")
    if ((abs(lambda.i.1-lambda.i)<Eps)|| (i>iMax) ) break
  }
  return(lambda.i.1)
}
```

Um Exemplo IV

```
# inicializar semente (opcional)
  set.seed(1000)
# gerar dados
  tam <- 10
  lambda <- 1.0
  y <- rpois(tam,lambda)
# tirar contagens nulas
  y <- y[y>0]
# testar funcao
  ajPoisson(y)
  Iter= 1  Lambda= 1.054764
  Iter= 2  Lambda= 0.9124157
  ...
  Iter= 14 Lambda= 0.7170033
  [1] 0.7170033
```

Fim

Questões? Envie-me um e-mail (fcruz@est.ufmg.br).

