

Introdução à Computação (MAC113 - FEA – 2020)

Ana C V de Melo

Comandos: Atribuição e Seleção

Sobre esta Aula

Os Programas

Comandos

Funções

O que já vimos

Sobre esta Aula

- ▶ tipos e estruturas de dados básicas:
 - ▶ vetores, listas, matrizes, data-frames

O que veremos hoje

- ▶ os primeiros comandos da linguagem:
 - ▶ atribuição e seleção
- ▶ como definimos e usamos funções? noções preliminares

Os Programas

Esquema de Programa

```
# preâmbulo - incluir elementos predefinidos que
# serão usados no programa

# funções: blocos de programas com parâmetros
imprime_valor <- function(x) {
  print(x)
}

# dados, por exemplo:
a <- 33 # variável a guarda o valor
b <- a + 22 # variável b usa valor de a
tabela <- read.csv("dados_exemplo.csv") # dados lidos

# sequência de comandos: transforma dados
if (a > 5) {
  imprime_valor("a é maior que 5")
}
imprime_valor(tabela)
```

Quem é o arquivo de dados?

- ▶ um arquivo *dados_exemplo.csv* ("Comma-separated values")

Produto	Preço
caneta azul	3.00
caneta vermelha	2.00
caderno 100fls	10.30
caderno 200fls	15.45

Execução de Programa

```
## [1] "a é maior que 5"

##          Produto Preço
## 1   caneta azul   3.00
## 2 caneta vermelha  2.00
## 3  caderno 100fls 10.30
## 4  caderno 200fls 15.45
```

Comandos Básicos

- ▶ Entrada e Saída (já visto)
- ▶ Atribuição (já visto)
- ▶ Seleção
- ▶ Repetição

Execução ...

```
Digite um número inteiro (e tecle enter): 12
[1] 12
[1] "integer"
Digite um número inteiro (e tecle enter): 23
[1] "23"
[1] "character"
Imprimindo valor de m: 23 e seu tipo: character
```

Atribuição

- ▶ Guarda valores em variáveis
- ▶ Variáveis: qq nome formado por letras, dígitos, "." e "_" (precisa começar com uma letra).

```
x <- 15
y = 22
z = x + y
print(z)
```

```
## [1] 37
```

Comandos

Entrada e Saída: teclado/tela

- ▶ Lê o que o usuário digita
- ▶ imprime na tela

```
n <- readline(prompt = " Digite um número inteiro (e tecle enter): ")
n <- as.integer(n)
print(n)
print(typeof(n))

m <- as.character(readline("Digite um número inteiro (e tecle enter): "))
print(m)
print(typeof(m))

cat("Imprimindo valor de m:", m,"e seu tipo:", typeof(m), "\n")
```

Entrada: arquivo texto

- ▶ Leitura de dados .csv

```
tabela <- read.csv(file= "dados_exemplo.csv",
                    header = TRUE, sep = ",") # leitura de dados
print(tabela)
```

```
##          Produto Preço
## 1   caneta azul   3.00
## 2 caneta vermelha  2.00
## 3  caderno 100fls 10.30
## 4  caderno 200fls 15.45
```

- ▶ já vimos que as planilhas são armazenadas como data-frames
- ▶ podemos manipular data-frames (vide aulas anteriores)

Problema: calcular média aritmética

- ▶ Dados 3 números inteiros calcular a média aritmética entre eles.
- ▶ Passo 1: planejar o que faremos no programa

```
# Programa
# Dados 3 números inteiros calcular a média aritmética entre eles

# lê os números

# calcula a média aritmética

# imprime o valor da média
```

Problema: calcular média aritmética

- ▶ Passo 2: fazer o programa

```
# Programa
# Dados 3 números inteiros calcular a média aritmética entre eles

# lê os números
a <- as.integer(readline("Digite o primeiro número: "))
b <- as.integer(readline("Digite o segundo número: "))
c <- as.integer(readline("Digite o terceiro número: "))

# calcula a media aritmética

# imprime o valor da média
```

```
Digite o primeiro número: 1
Digite o segundo número: 2
Digite o terceiro número: 3
```

Execução...

```
Digite o primeiro número: 34
Digite o segundo número: 23
Digite o terceiro número: 56
Média dos Números: 37.66667
```

Seleção

- ▶ If

```
if (condição) {
  sequência de comandos
}

ex:

x <- 5
if(x > 0){
  print("número positivo")
}
```

Problema: média e status do aluno

- ▶ Dadas 3 notas de um aluno (0 a 10) calcular a média aritmética e imprimir se o aluno foi: aprovado (≥ 5), reprovado (< 3) ou recuperação (≥ 3 e < 5).

Programa inteiro

```
# Programa
# Dados 3 números inteiros calcular a média aritmética entre eles

# lê os números
a <- as.integer(readline("Digite o primeiro número: "))
b <- as.integer(readline("Digite o segundo número: "))
c <- as.integer(readline("Digite o terceiro número: "))

# calcula a media aritmética

media <- (a + b + c) / 3

# imprime o valor da média

cat("Média dos Números:", media)
```

Problema: média e status do aluno

- ▶ Dadas 3 notas de um aluno (0 a 10) calcular a média aritmética e imprimir se o aluno foi: aprovado (≥ 5), reprovado (< 3) ou recuperação (≥ 3 e < 5).

Seleção

- ▶ If...else

```
if (condição) {
  sequência de comandos
}
else {
  sequência de comandos
}

ex:

x <- -5
if(x >= 0){
  print("Número não negativo")
}
else {
  print("Número negativo")
}

}
```

Programa

```
# lê os números
nota1 <- as.numeric(readline("Digite a primeira nota: "))
nota2 <- as.numeric(readline("Digite a segunda nota: "))
nota3 <- as.numeric(readline("Digite a terceira nota: "))

# calcula a media aritmética
media <- (nota1 + nota2 + nota3) / 3

# imprime
if (media >= 5){
  cat("Aprovado - Média: ", media)
}else{
  if (media < 3){
    cat("Reprovado - Média: ", media)
  }else{
    cat("Recuperação - Média: ", media)
  }
}
```

Execução ...

```
Digite a primeira nota: 4.5
Digite a segunda nota: 4.8
Digite a terceira nota: 5.8
Aprovado - Média: 5.033333
```

Funções

Problemas em códigos

- ▶ códigos longos são difíceis de ler
- ▶ muitas vezes repetimos partes do código em vários lugares: e se precisarmos mudar alguns comandos nessas partes dos códigos?
 - ▶ funções: uma noção já conhecida na matemática

Funções em R

- ▶ predefinidas: `read.csv()`, `readline()`, `print()`, `cat()`, `subset()`, ...
 - ▶ já **usamos** algumas! mas veremos muitas outras
 - ▶ são trechos de código que solucionam alguns problemas conhecidos
- ▶ definidas pelo usuário
 - ▶ temos novos problemas a serem resolvidos precisam ser organizados pelo usuário
 - ▶ podemos definir e usar novas funções

Funções definidas pelo usuário: sintaxe

- ▶ definição

```
<nome_da_função> <- function (<parametros>) {
  <sequência_de_comandos>
  return (<resultado>)
}
```

- ▶ *return* não é obrigatório

ex:

```
# função
# Dados 3 números inteiros calcular a média aritmética entre eles
CalculaMedia <- function(nota1, nota2, nota3){
  # calcula a média aritmética
  med <- (nota1 + nota2 + nota3) / 3

  # retorna o valor da média
  return(med)
}
```

O que acontece se executarmos esse código?

```
#definição da função
CalculaMedia <- function(nota1,nota2,nota3){
  # calcula a média aritmética
  med <- (nota1 + nota2 + nota3) / 3
  # retorna o valor da média
  return(med)
}
# Programa
# lê os números
nota1 <- as.numeric(readline("Digite a primeira nota: "))
nota2 <- as.numeric(readline("Digite a segunda nota: "))
nota3 <- as.numeric(readline("Digite a terceira nota: "))
# calcula a média aritmética
media <- CalculaMedia(nota1, nota2, nota3)
cat("Média: ", media)
```

- ▶ precisamos *chamar* (usar) a função

Execução

```
Digite a primeira nota: 6.7
Digite a segunda nota: 4.8
Digite a terceira nota: 5.8
Média: 5.766667
```

Mudando o problema

Fazer um script (programa) que:

- ▶ Define uma função (*CalculaMediaStatus*) para que dadas 3 notas de um aluno, calcula a média aritmética e retorna o status se o aluno foi: "aprovado" (≥ 5), "reprovado" (< 3) ou "recuperação" (≥ 3 e < 5).
- ▶ Lê uma planilha ("alunos.sub.csv") com dados dos alunos e suas provas: P1, P2 e Sub
- ▶ Usando a função definida, calcular o status dos 2 primeiros alunos da planilha

função

```
#definição da função
CalculaMediaStatus <- function(nota1,nota2,nota3){
# calcula a media aritmética
media <- (nota1 + nota2 + nota3) / 3
# retorna o valor do status
if (media >= 5){
  status <- "aprovado"
}else{
  if (media < 3){
    status <- "reprovado"
  }else{
    status <- "recuperação"
  }
}
return(status)
}
```

execução

```
Nome P1 P2 Sub
1 Alexandre 3.0 5.5 3.5
2 Carlos 5.0 4.5 6.5
3 Daniel 3.0 5.0 6.0
4 Tiago 6.7 8.0 4.8

Status dos Alunos
recuperação
aprovado
```

programa

```
# lê a planilha
alunos <- read.csv(file= "alunos.sub.csv",
                    header = TRUE, sep = ",") # leitura de dados
print(alunos)
# calcula a media aritmética
status1 <- CalculaMediaStatus(alunos[1,"P1"],
                              alunos[1,"P2"],
                              alunos[1,"Sub"])
status2 <- CalculaMediaStatus(alunos[2,"P1"],
                              alunos[2,"P2"],
                              alunos[2,"Sub"])
cat("\n Status dos Alunos \n", status1 ,"\n",
    status2)
```