

Introdução à Computação (MAC113 - FEA – 2020)

Ana C V de Melo

Programas, Valores, Tipos atômicos, Planilhas:
entrada/saída

Programas

Tipos de Dados Comuns: objetos

Programas

O que faz um programa?

- ▶ transforma um conjunto de dados de entrada em um conjunto de dados de saída
- ▶ precisamos então:
 - ▶ ler dados (entrada)
 - ▶ processar/transformar os dados
 - ▶ guardar/mostrar os resultados (dados de saída)
- ▶ quem são esses dados?
 - ▶ valores com tipos associados (ex.: números inteiros, números reais, "textos", etc.)

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")
- ▶ eles podem ser escritos como texto ou extraídos de uma planilha!

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
```

Tipos de Dados Comuns: objetos

Numéricos: inteiro (valor atômico)

```
i <- 20
print(typeof(i))      # typeof(.) dá o tipo do valor

## [1] "double"

j <- 20L
print(typeof(j))

## [1] "integer"
```

Numéricos: ponto flutuante (valor atômico)

```
r <- 20.23
print(typeof(r))

## [1] "double"
```

Operadores Aritméticos

##	Operador	Significado	Ex.
## 1	x + y	soma	2 + 3 = 5
## 2	x - y	subtração	2 - 3 = -1
## 3	x * y	multiplicação	2 * 3 = 6
## 4	x / y	divisão	2 / 3 = 0.6666667
## 5	x ^ y	exponenciação	2 ^ 3 = 8
## 6	x %% y	resto divisão	2 %% 3 = 2
## 7	x %/% y	divisão de inteiros	2 %/% 3 = 0

Valores Lógicos (atômicos)

```
l1 <- TRUE
l2 <- FALSE
print(typeof(l1))

## [1] "logical"

print(as.numeric(l1))  # valor numérico de l1

## [1] 1

print(as.numeric(l2))

## [1] 0

print(typeof(l1))      # veja que l1 permanece lógico

## [1] "logical"

► TRUE e FALSE_ são valores predefinidos - não são palavras!
```

Operadores Relacionais e Lógicos

Relacionais	Lógicos
x < y	!x
x > y	x y
x <= y	x && y
x >= y	
x == y	
x != y	

veremos o significado dos operadores lógicos mais tarde!!

Caracteres e strings

```
c1 <- 'a'
print(typeof(c1))
palavra <- "sim"
print(typeof(palavra))
frase <- "assistindo à aula de MAC0113"
print(typeof(frase))

## [1] "character"

## [1] "character"

## [1] "character"
```

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")
```

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
```

Entrada: planilhas de dados

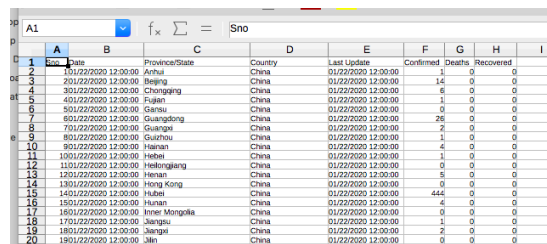
- podemos digitar um arquivo com os dados que desejamos

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

- podemos utilizar dados provenientes de planilhas ...

Entrada: dados de uma planilha

- planilha:



A1									
	A	B	C	D	E	F	G	H	I
1	Sno	Date	Province/State	Country	Last Update	Confirmed	Deaths	Recovered	
2	1	01/22/2020 12:00:00	Anhui	China	01/22/2020 12:00:00	1	0	0	
3	2	01/22/2020 12:00:00	Beijing	China	01/22/2020 12:00:00	14	0	0	
4	3	01/22/2020 12:00:00	Chongqing	China	01/22/2020 12:00:00	6	0	0	
5	4	01/22/2020 12:00:00	Fujian	China	01/22/2020 12:00:00	1	0	0	
6	5	01/22/2020 12:00:00	Gansu	China	01/22/2020 12:00:00	0	0	0	
7	6	01/22/2020 12:00:00	Guangdong	China	01/22/2020 12:00:00	26	0	0	
8	7	01/22/2020 12:00:00	Guangxi	China	01/22/2020 12:00:00	2	0	0	
9	8	01/22/2020 12:00:00	Guizhou	China	01/22/2020 12:00:00	1	0	0	
10	9	01/22/2020 12:00:00	Hainan	China	01/22/2020 12:00:00	4	0	0	
11	10	01/22/2020 12:00:00	Hebei	China	01/22/2020 12:00:00	1	0	0	
12	11	01/22/2020 12:00:00	Heilongjiang	China	01/22/2020 12:00:00	0	0	0	
13	12	01/22/2020 12:00:00	Henan	China	01/22/2020 12:00:00	5	0	0	
14	13	01/22/2020 12:00:00	Hong Kong	China	01/22/2020 12:00:00	0	0	0	
15	14	01/22/2020 12:00:00	Hubei	China	01/22/2020 12:00:00	444	0	0	
16	15	01/22/2020 12:00:00	Hunan	China	01/22/2020 12:00:00	4	0	0	
17	16	01/22/2020 12:00:00	Inner Mongolia	China	01/22/2020 12:00:00	0	0	0	
18	17	01/22/2020 12:00:00	Jiangsu	China	01/22/2020 12:00:00	1	0	0	
19	18	01/22/2020 12:00:00	Jiangxi	China	01/22/2020 12:00:00	2	0	0	
20	19	01/22/2020 12:00:00	Jilin	China	01/22/2020 12:00:00	0	0	0	

Figure 1: planilha

Como extrair os dados como .csv (comma-separated values)

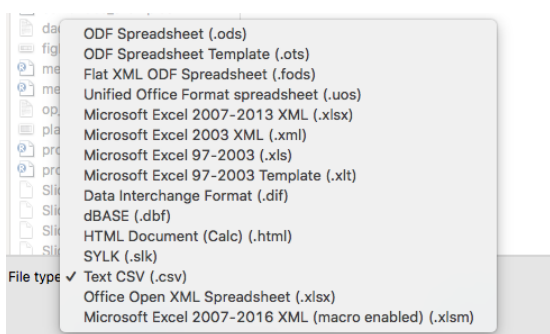
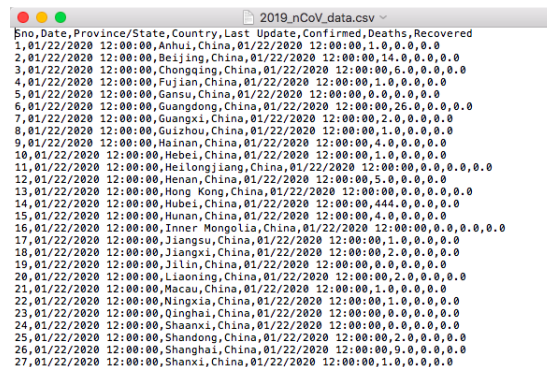


Figure 2: planilha.saveas

Planilha no formato .csv



Sno	Date	Province/State	Country	Last Update	Confirmed	Deaths	Recovered
1	01/22/2020 12:00:00	Anhui	China	01/22/2020 12:00:00	1	0	0
2	01/22/2020 12:00:00	Beijing	China	01/22/2020 12:00:00	14	0	0
3	01/22/2020 12:00:00	Chongqing	China	01/22/2020 12:00:00	6	0	0
4	01/22/2020 12:00:00	Fujian	China	01/22/2020 12:00:00	1	0	0
5	01/22/2020 12:00:00	Gansu	China	01/22/2020 12:00:00	0	0	0
6	01/22/2020 12:00:00	Guangdong	China	01/22/2020 12:00:00	26	0	0
7	01/22/2020 12:00:00	Guangxi	China	01/22/2020 12:00:00	2	0	0
8	01/22/2020 12:00:00	Guizhou	China	01/22/2020 12:00:00	1	0	0
9	01/22/2020 12:00:00	Hainan	China	01/22/2020 12:00:00	4	0	0
10	01/22/2020 12:00:00	Hebei	China	01/22/2020 12:00:00	1	0	0
11	01/22/2020 12:00:00	Heilongjiang	China	01/22/2020 12:00:00	0	0	0
12	01/22/2020 12:00:00	Henan	China	01/22/2020 12:00:00	5	0	0
13	01/22/2020 12:00:00	Hong Kong	China	01/22/2020 12:00:00	0	0	0
14	01/22/2020 12:00:00	Hubei	China	01/22/2020 12:00:00	444	0	0
15	01/22/2020 12:00:00	Hunan	China	01/22/2020 12:00:00	4	0	0
16	01/22/2020 12:00:00	Inner Mongolia	China	01/22/2020 12:00:00	0	0	0
17	01/22/2020 12:00:00	Jiangsu	China	01/22/2020 12:00:00	1	0	0
18	01/22/2020 12:00:00	Jiangxi	China	01/22/2020 12:00:00	2	0	0
19	01/22/2020 12:00:00	Jilin	China	01/22/2020 12:00:00	0	0	0
20	01/22/2020 12:00:00	Liaoning	China	01/22/2020 12:00:00	2	0	0
21	01/22/2020 12:00:00	Macau	China	01/22/2020 12:00:00	1	0	0
22	01/22/2020 12:00:00	Ningxia	China	01/22/2020 12:00:00	1	0	0
23	01/22/2020 12:00:00	Qinghai	China	01/22/2020 12:00:00	0	0	0
24	01/22/2020 12:00:00	Shaanxi	China	01/22/2020 12:00:00	0	0	0
25	01/22/2020 12:00:00	Shandong	China	01/22/2020 12:00:00	2	0	0
26	01/22/2020 12:00:00	Shanghai	China	01/22/2020 12:00:00	5	0	0
27	01/22/2020 12:00:00	Shanxi	China	01/22/2020 12:00:00	1	0	0

Figure 3: planilha.csv