MAC 113 — Introdução à Ciência da Computação

Aula 25

Nelson Lago

1°/2025

Previously on MAC113...

Comandos básicos com vectors e lists

- `vec <- 1:10`
- `vec <- seq_len(4)`
- `vec <- rep(3.14, 4)`
- `vec <- c("a", "b", "c")`
- `lista <- list(a=1, b=2, c=3)`
- `length(blah)` (list ou vector)
- `tudo_junto <- c(blah, bleh)` (lists ou vectors)
- `vec[X]`
- `lista[[X]]` (numérico)
- `blah[X:Y]` (numérico – lists ou vectors)
- `lista[[X]]` (character)
- `lista$X` (character, atalho sem aspas)
- `for (item in blah)` (list ou vector)
- `names(lista)`
  - `for (nome in names(lista))`
- `unlist(lista)`
- `vec[X] <- valor` (atribui)
- `vec <- append(vec, valor)` (acrescenta)
- `lista$nome <- valor` (atribui ou acrescenta)

Tabelas

Paciente	Colesterol (mg/dL)	Idade	Quarto	SUS
Fulano de Tal	164	49	132-A	S
Ciclano de Tal	227	83	231-B	N
Média/Total	195,6	66	–	1

- **Por que usamos tabelas?**
- **Provavelmente, várias razões** 😜
- **Uma delas é que se trata de uma visualização dupla**
- **Podemos olhar cada linha ou cada coluna**

Tabelas

- **Em tabelas desse tipo**
  - Cada linha corresponde a uma abstração (“pessoa”, “produto” etc.)
  - Cada coluna corresponde a uma coleção de valores de um mesmo tipo (“preço”, “colesterol”, “idade” etc.)
- **Podemos armazenar cada linha como uma list**
  - Mas processar as colunas (por exemplo, para calcular a média) é trabalhoso
- **Podemos armazenar cada coluna como um vector e usar os índices para identificar as linhas**
  - Mas já vimos antes que isso é um tanto trabalhoso também
- **Podemos usar uma list em que cada elemento é um vector correspondente a uma coluna**
  - Mas isso é apenas uma variação da ideia anterior

Precisamos de algo novo!

Algo que se assemelhe a uma tabela

Dataframes

(na verdade, tibbles)

Dataframes

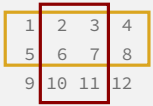
- **Dataframes são bastante similares a lists**
- **Cada item é um vector que representa uma coluna**

```
library(tibble)
#df <- data.frame(nome=c("Zé", "Fê", "Má", "Lú"), ID=c(1,2,3,4), nota=c(7, 8.5, 8.3, 9.1))
df <- tibble(nome=c("Zé", "Fê", "Má", "Lú"), ID=c(1,2,3,4), nota=c(7, 8.5, 8.3, 9.1))
cat("Alunos:", df$nome, "\n")
cat("Média da turma:", mean(df$nota), "\n")
cat("Lista de chamada:\n")
cat(paste(df$nome, " (", df$ID, ") ______", sep=""), sep="\n")
```

```
Alunos: Zé Fê Má Lú
Média da turma: 8.225
Lista de chamada:
Zé (1) ______
Fê (2) ______
Má (3) ______
Lú (4) ______
```

Recortes com dataframes

- **A maneira mais razoável de nos referirmos a elementos em uma tabela é através de suas coordenadas**



1	2	3	4
5	6	7	8
9	10	11	12

- **tabela**$_{l,c}$
- **tabela**$_{1-2,2-3}$

```
2 3
6 7
```

Dataframes

```r
library(tibble)
#df <- data.frame(nome=c("Zé", "Fê", "Má", "Lú"), ID=c(1,2,3,4), nota=c(7, 8.5, 8.3, 9.1))
df <- tibble(nome=c("Zé", "Fê", "Má", "Lú"), ID=c(1,2,3,4), nota=c(7, 8.5, 8.3, 9.1))
cat("Boletim de notas:\n")
for (i in seq_len(nrow(df))) {
  line <- df[i,]
  cat(unlist(line[c("nome", "nota")]), "\n")
}
```

```
Boletim de notas:
Zé 7
Fê 8.5
Má 8.3
Lú 9.1
```

Dataframes

- **Em geral, as operações vetoriais que fazemos em dataframes/tibbles processam as colunas**
  - Como cada coluna é um vector, sabemos que os elementos são todos do mesmo tipo
  - Em geral, coisas como `max()`, `mean()`, `sum()` etc., fazem sentido com as colunas
- **Para processar as linhas, podemos extrair uma linha específica por sua coordenada:**
  - `df[3,]`
- **ou, para processar todas, fazer um laço:**
  - `for (i in seq_len(nrow(df))) { df[i,] }`
- **ou usar `paste()`**
- **...**

Dataframes

Dataframes/tibbles: Como vivem? Onde moram? De que se alimentam?

- **Não tem muita graça digitar os dados de um dataframe/tibble como parte do programa**
- **O mais razoável é carregar de um arquivo**
- **Dataframes são tabelas $\longrightarrow$ o arquivo deve representar uma tabela**
- **O formato mais comumente usado é o CSV (comma-separated values)**

Nome	matemática	português	física	história
Alan Turing	9.7	1.4	9.2	8.7
Ada Lovelace	9.8	1.2	10.0	9.2

Dataframes

```
Nome,matemática,português,física,história
Alan Turing,9.7,1.4,9.2,8.7
Ada Lovelace,9.8,1.2,10.0,9.2
```

```
Nome          ,matemática,português,física,história
Alan Turing   ,9.7       ,1.4      ,9.2   ,8.7
Ada Lovelace  ,9.8       ,1.2      ,10.0  ,9.2
```

- CSV: “Comma” pode ser qualquer coisa (espaços, tabs...)

```
Nome;matemática;português;física;história
Alan Turing;9,7;1,4;9,2;8,7
Ada Lovelace;9,8;1,2;10,0;9,2
```

Dataframes

- **E como ler um arquivo desses?**

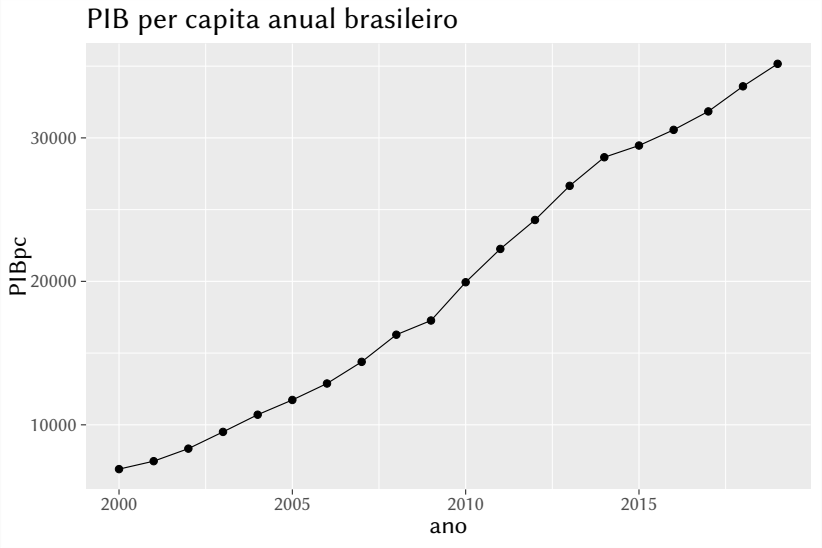
```
endereço <- 'http://www.ime.usp.br/~lago/dadinhos/renda_vs_inflacao.csv'
#df <- read.table(endereço, sep=",", header=TRUE) # dataframe
library(readr)
df <- read_delim(endereço, delim=",", col_types = "idd")
names(df)[2] <- "PIBpc"
print(head(df))
```

```
# A tibble: 6 × 3
    ano  PIBpc  IPCA
  <int>  <dbl> <dbl>
1  2000  6913.   7
2  2001  7467    6.8
3  2002  8341.   8.4
4  2003  9507.  14.7
5  2004 10706    6.6
6  2005 11734.   6.9
```

Dataframes

```
endereço <- 'http://www.ime.usp.br/~lago/dadinhos/renda_vs_inflacao.csv'
#df <- read.table(endereço, sep=",", header=TRUE) # dataframe
library(readr)
df <- read_delim(endereço, delim=",", col_types = "idd")
names(df)[2] <- "PIBpc"
p <- ggplot(df, aes(x=ano, y=PIBpc)) +
         geom_line() +
         geom_point() +
         labs(title="PIB per capita anual brasileiro")
print(p)
```

Dataframes


PIB per capita anual brasileiro
PIBpc
30000
20000
10000
2000
2005
2010
2015
ano

And now for something a little bit different

Recortes com filtros

- **Dado um dataframe `df` (ou um vector/lista), podemos fazer um "recorte" usando coordenadas: `df[linhas,colunas]`**
  - `linhas` e `colunas` podem ser valores únicos ou vectors
- **Também podemos usar um vector do tipo `LOGICAL`**

```
vec <- 1:5
cat(vec[c(TRUE, FALSE, FALSE, TRUE, FALSE)], "\n")
1 4
grandes <- vec > 3
cat(grandes, "\n")
FALSE FALSE FALSE TRUE TRUE
cat(vec[grandes], "\n")
4 5
cat(vec[vec > 3], "\n")
4 5
```

Recortes com filtros

```
library(tibble)
df <- tibble(nome=c("Fulano", "Ciclano", "Beltrano"), idade=c(17, 25, 19))
cat("Idade do Fulano:", unlist(df[df$nome == "Fulano","idade"]), "\n")
```

Idade do Fulano: 17

```
library(tibble)
df <- tibble(nome=c("Fulano", "Ciclano", "Beltrano"), idade=c(17, 25, 19))
cat("Maiores de idade:", unlist(df[df$idade>=18,]), "\n")
maiores <- df[df$idade>=18,]
cat("Maiores de idade:", paste(maiores$nome, ": ", maiores$idade, sep=""), sep="\n")
```

Maiores de idade: Ciclano Beltrano 25 19
Maiores de idade:
Ciclano: 25
Beltrano: 19

Exercício – dados de filmes

http://www.ime.usp.br/~lago/dadinhos/filmes.csv

nome	diretor	ano	nota	infantil	país
<chr>	<chr>	<int>	<dbl>	<chr>	<chr>
A hora mágica	Guilherme de A Prado	1999	6.7	N	Brasil
Central do Brasil	Walter Salles	1998	8	N	Brasil
Os incríveis	Brad Bird	2004	8	S	EUA
Na roda da fortuna	Joel Coen, Ethan Coen	1994	7.2	N	EUA
O fantasma do paraíso	Brian de Palma	1974	7.3	N	EUA
Asas do desejo	Wim Wenders	1987	7.9	N	Alemanha Ocidental
Peixe grande	Tim Burton	2003	8	N	EUA

Exercício – dados de filmes

- **Dada a lista de filmes em http://www.ime.usp.br/~lago/dadinhos/filmes.csv**
  - Quantos filmes há na lista?
  - Qual é a avaliação média dos filmes listados?
  - Qual o ano de lançamento mais antigo e o ano de lançamento mais novo da lista?
  - Qual é a mediana dos anos de lançamento?

```
library(readr)
main <- function() {
    arq <- 'http://www.ime.usp.br/~lago/dadinhos/filmes.csv'
    filmes <- read_delim(arq, delim=";", col_types = "ccidcc")
    cat("Há", nrow(filmes), "filmes na lista\n")
    cat("A avaliação média dos filmes é", mean(filmes$nota), "\n")
    cat("O filme mais antigo é de", min(filmes$ano),
        "e o filme mais novo é de", max(filmes$ano), "\n")
    cat("A mediana do ano de lançamento dos filmes é", median(filmes$ano), "\n")
}
main()
```

Exercício – dados de filmes

- **Dada a lista de filmes em http://www.ime.usp.br/~lago/dadinhos/filmes.csv**
  - Quantos filmes foram lançados no séc. XXI?

```
library(readr)
main <- function() {
    arq <- 'http://www.ime.usp.br/~lago/dadinhos/filmes.csv'
    filmes <- read_delim(arq, delim=";", col_types = "ccidcc")
    cat(nrow(filmes[filmes$ano > 2000,]), "filmes foram lançados no séc. XXI\n")
}
main()
```

Exercício – dados de filmes

- **Dada a lista de filmes em http://www.ime.usp.br/~lago/dadinhos/filmes.csv**
  - Quantos filmes foram lançados entre 1993 e 2007?

```
library(readr)
main <- function() {
    arq <- 'http://www.ime.usp.br/~lago/dadinhos/filmes.csv'
    filmes <- read_delim(arq, delim=";", col_types = "ccidcc")
    cat(nrow(filmes[filmes$ano >= 1993 & filmes$ano <= 2007,]),
        "filmes foram lançados entre 1993 e 2007\n")
}
main()
```

Exercício – dados de filmes

- **Dada a lista de filmes em http://www.ime.usp.br/~lago/dadinhos/filmes.csv**
  - Quais são os filmes infantis que há na lista?

```
library(readr)
main <- function() {
    arq <- 'http://www.ime.usp.br/~lago/dadinhos/filmes.csv'
    filmes <- read_delim(arq, delim=";", col_types = "ccidcc")
    print(filmes[filmes$infantil == "S",])
}
main()
```

mas pra que manter a coluna “infantil”?

Exercício – dados de filmes

- **Dada a lista de filmes em http://www.ime.usp.br/~lago/dadinhos/filmes.csv**
  - Quais são os filmes infantis que há na lista?

```
library(readr)
main <- function() {
    arq <- 'http://www.ime.usp.br/~lago/dadinhos/filmes.csv'
    filmes <- read_delim(arq, delim=";", col_types = "ccidcc")
    print(filmes[filmes$infantil == "S",names(filmes) != "infantil"])
}
main()
```

Intermezzo – o operador `%in%`

- Como saber se existe um elemento dentro de um vector?
- `if (valor %in% vector) { … }`

Exercício – dados de filmes

- **Dada a lista de filmes em http://www.ime.usp.br/~lago/dadinhos/filmes.csv**
  - Quais são os filmes infantis que há na lista?

```
library(readr)
main <- function() {
    arq <- 'http://www.ime.usp.br/~lago/dadinhos/filmes.csv'
    filmes <- read_delim(arq, delim=";", col_types = "ccidcc")
    filmes$infantil <- filmes$infantil == "S"
    print(filmes[filmes$infantil, ! names(filmes) %in% c("infantil", "país")])
}
main()
```

Exercício – dados de filmes

- **Dada a lista de filmes em http://www.ime.usp.br/~lago/dadinhos/filmes.csv**
  - Quais e quantos países diferentes aparecem na lista?

```
library(readr)
main <- function() {
    arq <- 'http://www.ime.usp.br/~lago/dadinhos/filmes.csv'
    filmes <- read_delim(arq, delim=";", col_types = "ccidcc")
    cat(unique(filmes$país), sep="\n")
    cat(length(unique(filmes$país)), "\n")
}
main()
```