

Ana C V de Melo

Tipos Compostos: data-frames

O que já vimos

Aulas Anteriores

- ▶ ...
- ▶ Tipos Compostos. Vetores e Listas: criar, acessar os elementos, modificar elementos, adicionar elementos, remover elementos.
- ▶ Matrizes: criar, acessar os elementos, modificar elementos, adicionar elementos, remover elementos

O que iremos ver hoje

Tipos Compostos:

- ▶ Data-frames: criar, acessar os elementos, modificar elementos, adicionar elementos, remover elementos
- ▶ como selecionar partes dos data-frames (`subset()`)

Data-frames

Lembrando Matrizes. . .

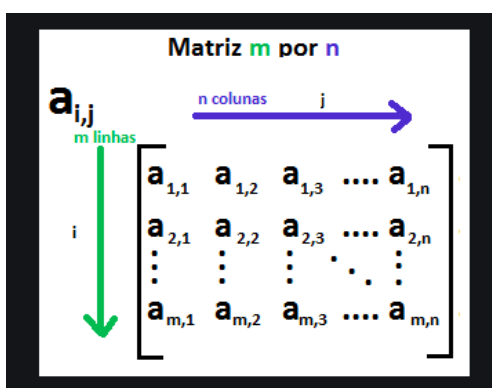


Figure 1:

Importância dos data-frames

- ▶ as planilhas são armazenadas como data-frames
- ▶ precisamos aprender a manipular data-frames
- ▶ eles são formados por listas
- ▶ estrutura bidimensional formada por listas

As estruturas que estamos aprendendo...

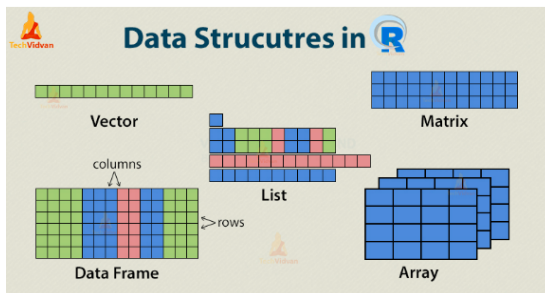


Figure 2:

Relembrando... - criar listas

```
v1 <- c(1,2,6,7:10)
v2 <- c(4,8,2)
l1 <- list(v1,v2) # criação de uma lista
print(l1)
```

```
## [[1]]
## [1] 1 2 6 7 8 9 10
##
## [[2]]
## [1] 4 8 2
```

► uma lista composta por 2 elementos: v1 e v2

Relembrando Listas - elementos com nomes

► nomes para cada um dos elementos na definição

```
l2 <- list(ce11= v1, ce12= v2)
print(l2)
```

```
## $ce11
## [1] 1 2 6 7 8 9 10
##
## $ce12
## [1] 4 8 2
```

Data-frames - criar

```
dados_alunos <- data.frame(
  Nome = c("Alexandre", "Allan", "Bernardo"),
  P1 = c(3.5, 5, 8, 9),
  P2 = c(5.5, 6, 7.5, 5.5),
  Sub = c(8.5, NA, NA, NA)
)
print(dados_alunos)
```

```
##      Nome  P1  P2 Sub
## 1 Alexandre 3.5 5.5 8.5
## 2      Allan 5.0 6.0  NA
## 3      Andre 8.0 7.5  NA
## 4    Bernardo 9.0 5.5  NA
```

```
typeof(dados_alunos)
```

```
## [1] "list"
```

Acesso aos elementos dos data-frames - 1 elemento

```
print(dados_alunos)
```

```
##      Nome  P1  P2 Sub
## 1 Alexandre 3.5 5.5 8.5
## 2      Allan 5.0 6.0  NA
## 3      Andre 8.0 7.5  NA
## 4    Bernardo 9.0 5.5  NA
```

```
print(dados_alunos[1,2]) # linha 1, coluna 2
```

```
## [1] 3.5
```

```
print(dados_alunos[1,"P1"]) # usando o nome da coluna
```

```
## [1] 3.5
```

Acesso aos elementos dos data-frames - colunas

```
print(dados_alunos)
```

```
##      Nome  P1  P2 Sub
## 1 Alexandre 3.5 5.5 8.5
## 2      Allan 5.0 6.0  NA
## 3      Andre 8.0 7.5  NA
## 4    Bernardo 9.0 5.5  NA
```

```
print(dados_alunos["Nome"]) # coluna "Nome"
```

```
##      Nome
## 1 Alexandre
## 2      Allan
## 3      Andre
## 4    Bernardo
```

► o data-frame apenas com a coluna "Nome"

Acesso aos elementos dos data-frames - colunas

```
print(dados_alunos)
```

```
##      Nome  P1  P2 Sub
## 1 Alexandre 3.5 5.5 8.5
## 2      Allan 5.0 6.0  NA
## 3      Andre 8.0 7.5  NA
## 4    Bernardo 9.0 5.5  NA
```

```
print(dados_alunos$Nome) # coluna "Nome"
```

```
## [1] Alexandre Allan      Andre      Bernardo
## Levels: Alexandre Allan Andre Bernardo
```

► resulta em um vetor com os elementos da coluna "Nome"

Acesso aos elementos dos data-frames - colunas

```
print(dados_alunos)
```

```
##      Nome  P1  P2 Sub
## 1 Alexandre 3.5 5.5 8.5
## 2      Allan 5.0 6.0  NA
## 3      Andre 8.0 7.5  NA
## 4    Bernardo 9.0 5.5  NA
```

```
print(dados_alunos[1:3]) # da col 1 até a col 3
```

```
##      Nome  P1  P2
## 1 Alexandre 3.5 5.5
## 2      Allan 5.0 6.0
## 3      Andre 8.0 7.5
## 4    Bernardo 9.0 5.5
```

Acesso aos elementos dos data-frames - linhas

```
print(dados_alunos)

##      Nome  P1  P2 Sub
## 1 Alexandre 3.5 5.5 8.5
## 2      Allan 5.0 6.0  NA
## 3      Andre 8.0 7.5  NA
## 4    Bernardo 9.0 5.5  NA
```

```
print(dados_alunos[1, ]) # 1a linha
```

```
##      Nome  P1  P2 Sub
## 1 Alexandre 3.5 5.5 8.5
```

Acesso aos elementos dos data-frames - linhas

```
print(dados_alunos)
```

```
##      Nome  P1  P2 Sub
## 1 Alexandre 3.5 5.5 8.5
## 2      Allan 5.0 6.0  NA
## 3      Andre 8.0 7.5  NA
## 4    Bernardo 9.0 5.5  NA
```

```
print(dados_alunos[2:4, ]) # da 2a até a 4a linha
```

```
##      Nome  P1  P2 Sub
## 2      Allan 5 6.0  NA
## 3      Andre 8 7.5  NA
## 4    Bernardo 9 5.5  NA
```

Qual o tipo associado aos data-frames?

```
typeof(dados_alunos) # todo o data-frame
```

```
## [1] "list"
```

```
typeof(dados_alunos[1, ]) # 1 linha
```

```
## [1] "list"
```

```
typeof(dados_alunos$P1) # 1 coluna
```

```
## [1] "double"
```

Tipos: linhas X colunas

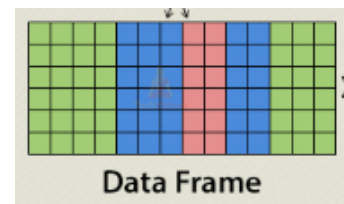


Figure 3:

- ▶ a grosso modo, data-frames são listas (linhas) de vetores (colunas)!

Criar um data-frame a partir de um existente

- ▶ seleciona apenas algumas colunas - por nomes
- ▶ o que acontece quando selecionamos uma coluna por nome?

```
coluna_nome <- dados_alunos$Nome
coluna_P2 <- dados_alunos$P2
print(coluna_nome)
```

```
## [1] Alexandre Allan      Andre      Bernardo
## Levels: Alexandre Allan Andre Bernardo
```

```
print(coluna_P2)
```

```
## [1] 5.5 6.0 7.5 5.5
```

- ▶ teremos 2 vetores com os valores das colunas

Criar um data-frame a partir de um existente

- ▶ seleciona apenas algumas colunas - por nomes
- ▶ precisamos da estrutura data-frame

```
dados_alunos1 <- data.frame(dados_alunos$Nome,
                             dados_alunos$P2)
print(dados_alunos1) #cols Nome e P2
```

```
##      dados_alunos.Nome  dados_alunos.P2
## 1      Alexandre      5.5
## 2      Allan      6.0
## 3      Andre      7.5
## 4      Bernardo      5.5
```

Criar um data-frame a partir de um existente

- ▶ seleciona apenas algumas colunas - por índices

```
dados_alunos1 <- dados_alunos[1:3] # seleciona colunas
print(dados_alunos1) #cols de 1 a 3
```

```
##      Nome  P1  P2
## 1 Alexandre 3.5 5.5
## 2      Allan 5.0 6.0
## 3      Andre 8.0 7.5
## 4    Bernardo 9.0 5.5
```

Criar um data-frame a partir de um existente

- ▶ seleciona apenas algumas colunas - por índices
- ▶ mesmo efeito do anterior - notação parecida com matriz

```
dados_alunos1 <- dados_alunos[,1:3] # todas as linhas
print(dados_alunos1) #cols de 1 a 3
```

```
##      Nome  P1  P2
## 1 Alexandre 3.5 5.5
## 2      Allan 5.0 6.0
## 3      Andre 8.0 7.5
## 4    Bernardo 9.0 5.5
```

Criar um data-frame a partir de um existente

- ▶ seleciona algumas linhas - mesma ideia de matrizes

```
dados_alunos2 <- dados_alunos[c(1,3), ] # linhas 1 e 3
print(dados_alunos2)                   # todas as colunas
```

```
##      Nome  P1  P2 Sub
## 1 Alexandre 3.5 5.5 8.5
## 3      Andre 8.0 7.5  NA
```

Criar um data-frame a partir de um existente

- ▶ seleciona algumas linhas e algumas colunas

```
dados_alunos2 <- dados_alunos[c(1,3), c(1:3)]
print(dados_alunos2) # linhas 1 e 3, cols 1 a 3
```

```
##      Nome  P1  P2
## 1 Alexandre 3.5 5.5
## 3      Andre 8.0 7.5
```

Adicionar linhas a um data-frame - 1

- cria um novo data-frame com 1 ou mais linhas

```
dados_alunos_novos <- data.frame(
  Nome = c("Bruna", "Camila"),
  P1 = c(9, 6.5),
  P2 = c(4.6, 7.3)
)

print(dados_alunos_novos)
```

```
##      Nome  P1  P2
## 1 Bruna 9.0 4.6
## 2 Camila 6.5 7.3
```

Adicionar linhas a um data-frame - 2

- ▶ adiciona ao final do data-frame existente

```
print(dados_alunos2)
```

```
##      Nome  P1  P2
## 1 Alexandre 3.5 5.5
## 3      Andre 8.0 7.5
```

```
dados_alunos3 <- rbind(dados_alunos2, dados_alunos_novos)
print(dados_alunos3)
```

```
##      Nome  P1  P2
## 1 Alexandre 3.5 5.5
## 3      Andre 8.0 7.5
## 11 Bruna 9.0 4.6
## 2 Camila 6.5 7.3
```

Adicionar linhas a um data-frame

- ▶ adiciona no início do data-frame existente - veja a posição dos dados

```
print(dados_alunos2)
```

```
##      Nome  P1  P2
## 1 Alexandre 3.5 5.5
## 3      Andre 8.0 7.5
```

```
dados_alunos4 <- rbind(dados_alunos_novos, dados_alunos2)
print(dados_alunos4)
```

```
##      Nome  P1  P2
## 1 Bruna 9.0 4.6
## 2 Camila 6.5 7.3
## 11 Alexandre 3.5 5.5
## 3      Andre 8.0 7.5
```

Adicionar coluna ao data-frame

- ▶ cria um data-frame com a nova coluna e adiciona ao data-frame existente

```
dados_colunaSub <- data.frame(Sub=c(6.5, NA, 5.0, NA))
dados_alunos3 <- cbind(dados_alunos3, dados_colunaSub)
print(dados_alunos3)
```

```
##      Nome  P1  P2 Sub
## 1 Alexandre 3.5 5.5 6.5
## 3      Andre 8.0 7.5  NA
## 11 Bruna 9.0 4.6 5.0
## 2 Camila 6.5 7.3  NA
```

- ▶ NA - Not Available

Adicionar coluna ao data-frame

- ▶ cria uma nova coluna com nome "Rec" e os respectivos valores

```
dados_alunos3$Rec <- c(8.5, NA, 8.5, NA)
print(dados_alunos3)
```

```
##      Nome  P1  P2 Sub Rec
## 1 Alexandre 3.5 5.5 6.5 8.5
## 3      Andre 8.0 7.5  NA  NA
## 11 Bruna 9.0 4.6 5.0 8.5
## 2 Camila 6.5 7.3  NA  NA
```

Mais operações sobre data-frames - seleção por valores

- ▶ alunos com nota maior ou igual a 8 na P1
- ▶ todas as linhas que satisfazem essa condição

```
print(dados_alunos3)
```

```
##      Nome  P1  P2 Sub Rec
## 1 Alexandre 3.5 5.5 6.5 8.5
## 3      Andre 8.0 7.5  NA  NA
## 11 Bruna 9.0 4.6 5.0 8.5
## 2 Camila 6.5 7.3  NA  NA
```

```
alunos_p1 <- subset(dados_alunos3, dados_alunos3$P1 >= 8)
print(alunos_p1)
```

```
##      Nome  P1  P2 Sub Rec
## 3      Andre 8 7.5  NA  NA
## 11 Bruna 9 4.6 5 8.5
```

Mais operações sobre data-frames - seleção por valores

- ▶ alunos que Não (!) obtiveram nota maior ou igual a 8 na P1

```
print(dados_alunos3)
```

```
##      Nome  P1  P2 Sub Rec
## 1 Alexandre 3.5 5.5 6.5 8.5
## 3      Andre 8.0 7.5  NA  NA
## 11     Bruna 9.0 4.6 5.0 8.5
## 2      Camila 6.5 7.3  NA  NA
```

```
alunos_p1<-subset(dados_alunos3, !(dados_alunos3$P1 >= 8))
print(alunos_p1)
```

```
##      Nome  P1  P2 Sub Rec
## 1 Alexandre 3.5 5.5 6.5 8.5
## 2      Camila 6.5 7.3  NA  NA
```

Mais operações sobre data-frames - seleção por valores

- ▶ alunos com nota menor que 5 ou (!) maior que 8 na P1

```
print(dados_alunos3)
```

```
##      Nome  P1  P2 Sub Rec
## 1 Alexandre 3.5 5.5 6.5 8.5
## 3      Andre 8.0 7.5  NA  NA
## 11     Bruna 9.0 4.6 5.0 8.5
## 2      Camila 6.5 7.3  NA  NA
```

```
alunos_p1<- subset(dados_alunos3, dados_alunos3$P1 < 5 |
                  dados_alunos3$P1 > 8)
print(alunos_p1)
```

```
##      Nome  P1  P2 Sub Rec
## 1 Alexandre 3.5 5.5 6.5 8.5
## 11     Bruna 9.0 4.6 5.0 8.5
```

Mais operações sobre data-frames - seleção por valores

- ▶ alunos com nota entre 3 e (&) 7 na P1

```
print(dados_alunos3)
```

```
##      Nome  P1  P2 Sub Rec
## 1 Alexandre 3.5 5.5 6.5 8.5
## 3      Andre 8.0 7.5  NA  NA
## 11     Bruna 9.0 4.6 5.0 8.5
## 2      Camila 6.5 7.3  NA  NA
```

```
alunos_p1<- subset(dados_alunos3, dados_alunos3$P1 >= 3 &
                  dados_alunos3$P1 <= 7)
```

```
print(alunos_p1)
```

```
##      Nome  P1  P2 Sub Rec
## 1 Alexandre 3.5 5.5 6.5 8.5
## 2      Camila 6.5 7.3  NA  NA
```

Como remover linhas ou colunas?

- ▶ já vimos como fazer essas operação para matrizes
- ▶ como você acha que podemos fazer isso para os data-frames? experimente!