

CURSO R - 2023



—

NAIARA ALCANTARA



TÓPICOS

1-INTRODUÇÃO AO R

2- ANÁLISE EXPLORATÓRIA E
MANIPULAÇÃO DOS DADOS

3- SALVAMENTO E ABERTURA

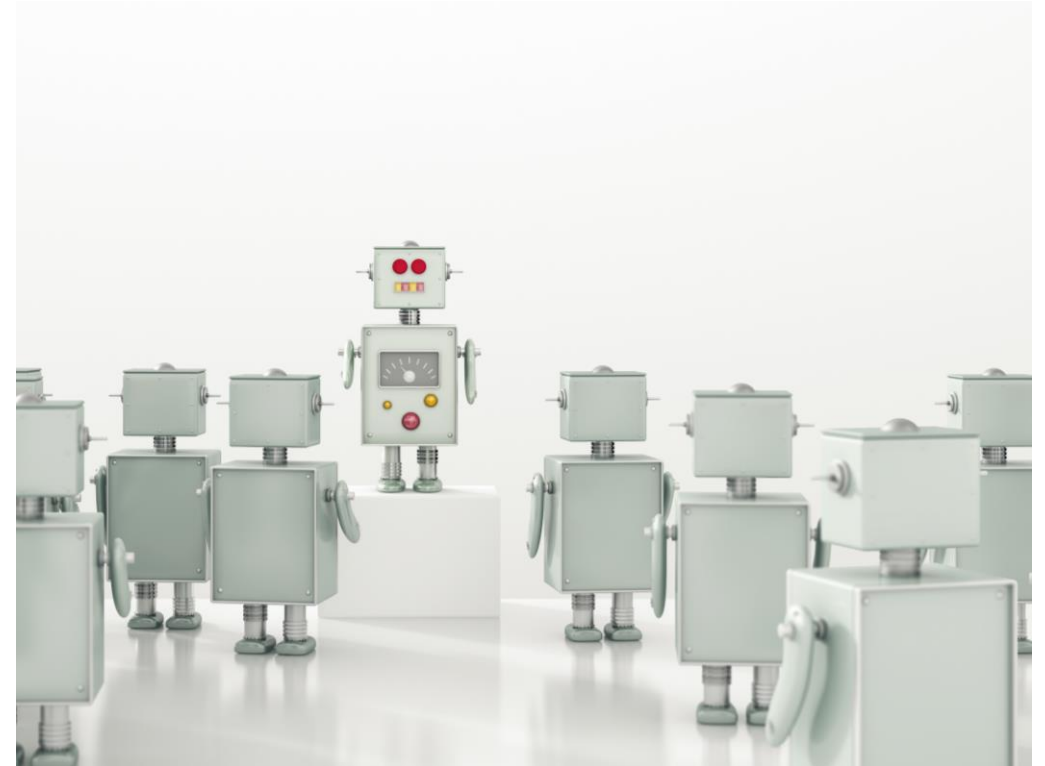
4- ANÁLISES DESCRITIVAS

5- APRESENTAÇÃO GRÁFICA

6- PROCESSAMENTO DE DADOS

7- ANÁLISES INFERENCIAIS

- Funções para observar a estrutura dos dados
- Seleção
- Exclusões
- Inserções e filtros
- Alterações
- Tipos de variáveis





ANÁLISE EXPLORATÓRIA E ORGANIZAÇÃO DOS DADOS

ANÁLISE DO DATA FRAME

Estrutura dos dados

```
#Função para análise das classes de cada variável|
str(Ficha_Pacientes)
# 'data.frame': 4 obs. of 7 variables:
#  $ Id          : int  1 2 3 4
#  $ Nomes       : chr  "Ana" "Gilberto" "Rodrigo" "Marcela"
#  $ Peso        : num  75.6 99 62.8 102
#  $ Idades      : num  25 18 44 23
#  $ Escolaridade : chr  "Graduação" "Mestrado" "Primário" "Graduação"
#  $ Exerc_Recomend : chr  "Natação" "Pilates" "Musculação" "Corrida"
#  $ Comidas_preferidas: chr  "Chocolate" "Sorvete" "Milho" "Pão"
```

- número, valor real, *numeric*, *double*
- texto, string, *character*, caracteres
- lógico, *logical*, booleano, valor TRUE/FALSE

Int são os números inteiros

- O R identifica o tipo de variável e diz qual seu tipo.
- É possível que o tipo não esteja adequado, nesse caso é necessário alterar o tipo usando a função “as.” com o tipo que queremos que seja inserido na variável, e depois para verificar se o tipo está correto usaremos a função “is.”. Falaremos mais sobre isso, em uma etapa mais avançada do curso

```
#A base de dados deverá ter um nome de coluna e linhas
#Para saber esses nomes use as funções rownames e colnames|
rownames(Ficha_Pacientes)
#[1] "1" "2" "3" "4"
colnames(Ficha_Pacientes)#Usando a função names também
#Conseguimos ver todas as colunas
# [1] "Id"          "Nomes"          "Peso"          "Idades"
# [5] "Escolaridade" "Exerc_Recomend" "Comidas_preferidas"
```

ANÁLISE DO DATA FRAME - SELEÇÕES

```
#Análise das médias
print(mean(Ficha_Pacientes$Idades))
#[1] 27.5
summary(Ficha_Pacientes$Idades)
# Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# 18.00  21.75   24.00   27.50  29.75   44.00
summary(Ficha_Pacientes$Exerc_Recomend)
# Length      Class      Mode
#  4 character character
table(Ficha_Pacientes$Escolaridade)
# Graduação  Mestrado  Primário
#  2          1        1
```



O cifrão é um caractere importante, pois permite que selecionemos um objeto dentro de outro.

Permite chamar uma variável dentro de uma base específica

```
#Seleção dos dados indicando 1º a linha e 2º a coluna
Ficha_Pacientes[2,5]
#[1] "Mestrado"
Ficha_Pacientes[2,5, drop=F]
# Escolaridade
# 2      Mestrado
Ficha_Pacientes[2,c("Peso", "Idades", "Nomes")]
# Peso Idades    Nomes
# 2    99      18 Gilberto
```

MANIPULAÇÃO DO DATA FRAME - EXCLUSÕES

```
#Exclusão da primeira linha
Ficha_Pacientes[-1, ]
#Exclusão da primeira linha e última coluna
Ficha_Pacientes[ -1 , -7]
```



Cadê a Ana?

	Id	Nomes	Peso	Idades	Escolaridade	Exerc_Recomend	Comidas_preferidas
2	2	Gilberto	99.0	18	Mestrado	Pilates	Sorvete
3	3	Rodrigo	62.8	44	Primário	Musculação	Milho
4	4	Marcela	102.0	23	Graduação	Corrida	Pão

! Lembre-se que os resultados somente estão impressos na base, mas não estão salvos

```
#Exclusões de colunas
## deleta quant_filhos
Ficha_Pacientes$Quant_filhos ← NULL

## deleta colunas específicas,
#0 mesmo pode ser feito para linhas
Ficha_Pacientes ← Ficha_Pacientes[, c(-4, -6)]
```

! Todas essas inclusões e exclusões serão muito importante na manipulação real de dados.

MANIPULAÇÃO DO DATA FRAME - INSERÇÕES E FILTROS

```
#Inserir novos dados em toda a base
Ficha_Pacientes$Sexo ← c("F", "M", "M", "F")
Ficha_Pacientes$Quant_filhos ← c(4:7)
```

```
#Uma outra forma de inserção de dados
#é utilizando uma função muito importante no rbase
# a função cbind
```

```
Ficha_Pacientes ← cbind(Ficha_Pacientes,
                          Prim_emplo =
                          c("sim", "nao", "nao", "sim"))
```



O subset é uma forma de selecionar e criar bases menores, inserindo somente suas variáveis de interesse- **Rbase**

FILTRAR USANDO O SUBSET = SUBCONJUNTO

```
#Subset:
#Quero uma base de dados com apenas 5 variáveis
subset(Ficha_Pacientes, select = c("Id", "Nomes", "Peso",
                                   "Idades", "Sexo"))

Base_menor ← subset(Ficha_Pacientes, select =
                    c("Id", "Nomes", "Peso",
                      "Idades", "Sexo"))
```

```
#Mas eu queria apenas pessoas do sexo masculino
Base_menor ← subset(Ficha_Pacientes,
                    Sexo == "M",
                    select = c("Id", "Nomes", "Peso", "Idades", "Sexo"))

#Mas eu tbém queria pessoas com idades maiores ou
#igual a 20 anos
Base_menor ← subset(Ficha_Pacientes,
                    Idades ≥ 20,
                    select = c("Id", "Nomes", "Peso", "Idades", "Sexo"))
```

MANIPULAÇÃO DO DATA FRAME –FILTROS E ALTERAÇÕES

```
#Filtro usando o filter
# seleciona apenas colunas numéricas
Filter(is.numeric, Ficha_Pacientes)
```

```
# Id  Peso  Idades
# 1   1   75.6     25
# 2   2   99.0     18
# 3   3   62.8     44
# 4   4  102.0     23
```

```
# seleciona apenas colunas de texto
```

```
Filter(is.character, Ficha_Pacientes)
```

```
#      Nomes Escolaridade Exerc_Recomend Comidas_preferidas
# 1      Ana   Graduação      Natação      Chocolate
# 2 Gilberto   Mestrado      Pilates      Sorvete
# 3 Rodrigo   Primário   Musculação      Milho
# 4  Marcela   Graduação      Corrida      p=
```

```
#Separando dados das colunas####
```

```
library(stringr)
```

```
#Inserindo uma variável com 2 informações
```

```
Ficha_Pacientes$Nome_Mãe <-
```

```
c('John, Mae', 'Maude, Lebowski', 'Mia, Amy', 'Andy, James')
```

```
#Separando
```

```
str_split_fixed(Ficha_Pacientes$Nome_Mãe, " ", 2)
```

```
#salvando na base |
```

```
Ficha_Pacientes$Nomes <-
```

```
str_split_fixed(Ficha_Pacientes$Nome_Mãe, " ", 2)
```


MANIPULAÇÃO DO DATA FRAME -ALTERAÇÕES

Alterando valores individuais

```
#Vamos alterar a idade da Marcela de 23 para 53
Ficha_Pacientes$Idades[4] ← 53
#Seguindo essa mesma lógica podemos alterar o nome das variáveis
#Ou então criar uma cópia com um nome diferente
```

Alterando grandes conjuntos de dados

```
library(memisc)

#Quero transformar a idade, fazer com que ela deixe de
#Ser numérica e se torne categorica (faixas de idade)
Ficha_Pacientes$Idade_Faixas ← recode(Ficha_Pacientes$Idades,
"Jovens" ← c(18:25), "Idoso" ← 44)
#Assim podemos alterar qualquer variável, mas é importante
#lembrar que tipo de variável estamos alterando
```

! Antes de usar o recode
pense no tipo de variável
que está sendo
modificada

MANIPULAÇÃO DO DATA FRAME – TIPOS DE VARIÁVEIS NO AMBIENTE DE PROGRAMAÇÃO

- No slide 14 (clique aqui) – [naiaara](#) colocar o link, nós vimos que existem tipos diferentes de variáveis, essa informação é essencial em várias etapas da análise dos dados, desde a construção do data frame, até a utilização de testes mais robustos estatisticamente.
- Nesse momento veremos a importância dos tipos das variáveis para análise dos dados e recodificação
- No ambiente de programação do R, os tipos de variáveis recebem nomes em inglês, os principais são:

FACTOR são um tipo de dado especial usado principalmente para representar variáveis categóricas.

Quando você especifica um objeto como um fator, automaticamente o R irá atribuir uma ordem para esses dados

Numeric		Factor	
Character	Integer		
Nomes	Peso	Idades	Escolaridade
Gilberto	99.0	18	Mestrado
Marcela	102.0	23	Graduação
Ana	75.6	25	Graduação
Rodrigo	62.8	44	Primário

MANIPULAÇÃO DO DATA FRAME - TIPOS DE VARIÁVEIS NO AMBIENTE DE PROGRAMAÇÃO

```
#Podemos alterar qualquer variável, mas é importante  
#lembrar que tipo de variável estamos alterando  
#e entender como essa variável estará após as alterações
```

```
Ficha_Pacientes$Escolaridade ←  
  as.factor(Ficha_Pacientes$Escolaridade)
```

```
#Olhar a estrutura dos dados  
str(Ficha_Pacientes)
```

```
#Verificar os níveis  
Ficha_Pacientes$Escolaridade  
#Levels: Graduação Mestrado Primário
```

```
#Se eu não concordar com a ordem atribuída pelo R,  
#Se essa ordem não fizer sentido para as análises estatísticas  
#então eu posso alterar a ordem usando a função levels  
Ficha_Pacientes$Escolaridade2 ←  
  factor(Ficha_Pacientes$Escolaridade,  
  levels = c("Primário", "Graduação", "Mestrado"))
```

```
#as.numeric/as.character####  
#Em geral o ambiente de programação entende quais são  
#os tipos das variáveis, porém pode ser que ele entenda errado  
#ou que em razão da abertura da base de dados seja necessário  
#fazer alguns ajustes, dizendo para o ambiente o que é numérico  
#e o que é caracter, nesses casos usamos as funções as. =como
```

as.character

as.numeric



OBRIGADO

NAIARA ALCANTARA
NAYARA_SANDY@HOTMAIL.COM