I Seminário de Extensão do Programa UFMS Digital

Estatística Descritiva no Software R

Trilha Temática IV – Ciência de Dados

Data: 31/05/2024

Dra. Magaly del Carmen Fonseca Medrano

magaly.fonseca@ufms.br

EaD UFMS Digital Graduação

AGEAD Agência de Educação Digital e a Distância

UFMS

Programação

1. Introdução ao R (30min)

1.1 Definições e Conceitos Técnicos de Programação

1.2 Apresentação geral do ambiente e Instalação

1.3 Noções de Sintaxe e Estrutura de funções

1.4 Comandos Básicos

1.5 Instalação de Pacotes

2. Manipulação de Dados com R (30min)

2.1 Tipos de Estrutura (Vetores, Data frames e Matrizes e Listas)

2.2 Importação e exportação de dados

3. Estatística Descritiva (30min)

3.1 Medidas de Tendencia Central

3.2 Medidas de Variabilidade

3.3 Medidas de Posição

3.4 Layout de Gráficos

1. Introdução ao R (30min)

1.1 Definições e Conceitos Técnicos de Programação

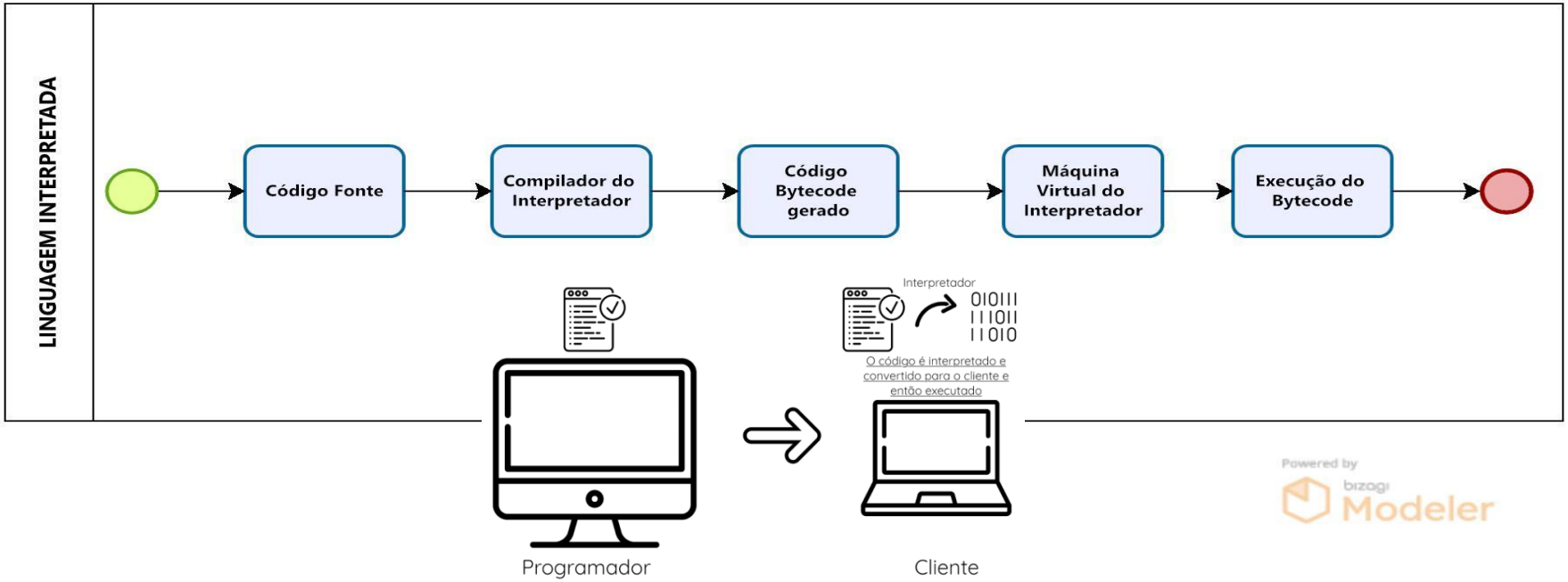
1.2 Apresentação geral do ambiente e Instalação

1.3 Noções de Sintaxe e Estrutura de funções

1.4 Comandos Básicos

1.5 Instalação de Pacotes

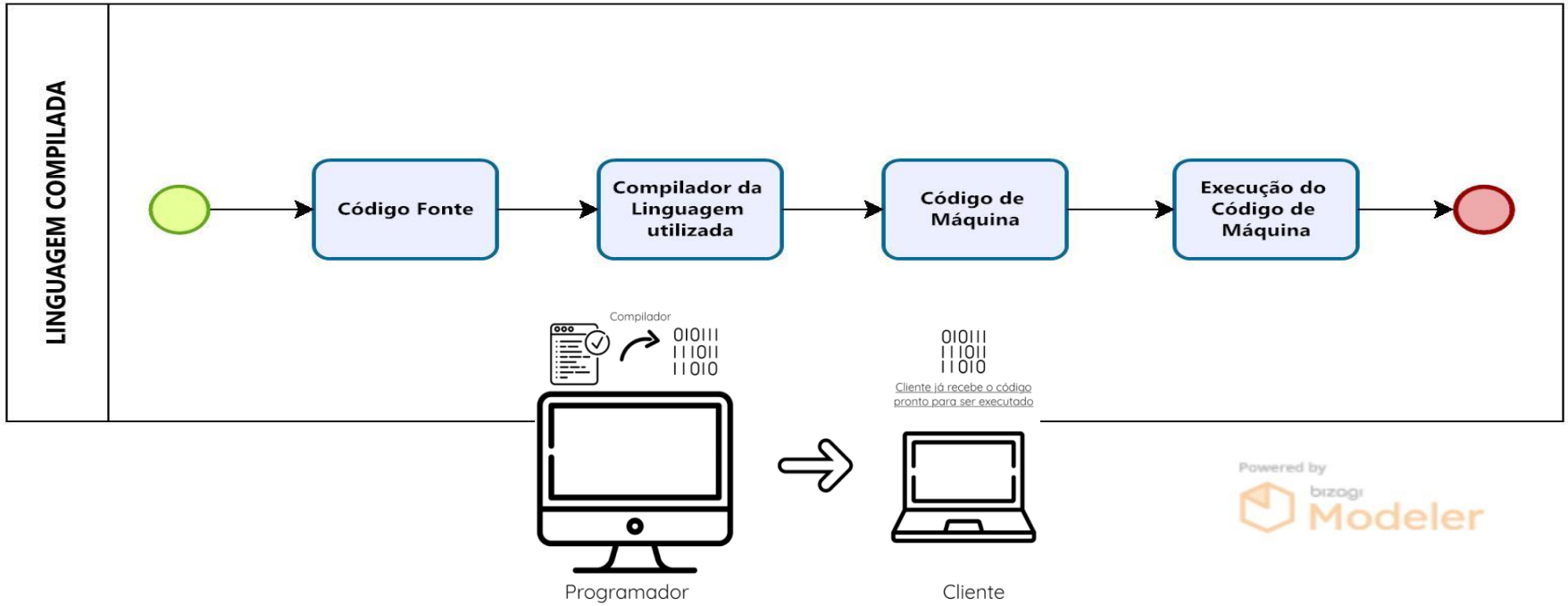
Linguagem de programação interpretada – Alto



Fonte: Hcode(2024); Rosa(2021)

1.1 Definições e Conceitos Técnicos de Programação

Linguagem de programação



Fonte: Hcode(2024); Rosa(2021)

1.1 Definições e Conceitos Técnicos de Programação

Aspecto	Linguagens Interpretadas	Avaliação	Linguagens Compiladas	Avaliação	Linguagens Híbridas	Avaliação
Disponibilidade de Código Fonte	Público	+	Privado	−	Público/Privado	+ −
Execução de processos	Execução mais lenta	−	Execução rápida	+	Lenta/Rápida	− +
Depuração de Código	Fácil depuração	+	Otimização geral	+ +	Fácil e Completa	+ + +
Compilação dos processos	Gera Bytecode	+	Gera Código de Máquina	−	Requer compilação em algumas partes	+ −
Portabilidade	Mais portátil, executado por uma máquina virtual.	+	Menos portátil, requer compilação específica para cada plataforma.	−	Portabilidade moderada, pode ser compilado ou interpretado	+ −
Segurança	O código-fonte é visível e pode ser facilmente inspecionado	−	O código de máquina é mais difícil de entender.	+	Nível de segurança variável	− +
Facilidade de Manutenção	Facil atualização e manutenção	+	Desafiador devido à necessidade de gerenciar diferentes versões compiladas	−	Flexibilidade de atualizações	+ −
Exemplos de Linguagens	R, Python, JavaScript, Ruby, PHP, etc.	3	C, C++, Go, Rust, Swift, etc.	0	Java, Kotlin, Scala, C,C++, etc.	3

Compiled vs. Interpreted Languages (youtube.com)

1.1 Definições e Conceitos Técnicos de Programação

Desenvolvedores/Programadores

- Criam pacotes R
- Possui habilidades avançadas em programação e conhecimento profundo da linguagem R
- Podem criar novas funções, algoritmos ou ferramentas, encapsulá-los em pacotes e disponibilizá-los para a comunidade R

Linguagem de Programação Interpretada com abordagem híbrida

Usuários

- Utilizam o R para análise de dados, estatísticas, visualizações e outras tarefas relacionadas
- Podem não ter uma formação em programação, mas estão familiarizados com a linguagem R

1.2 Apresentação Geral do Ambiente e Instalação

O que é o software R?

O **R** é uma linguagem e ambiente estatístico que traz muitas vantagens para o usuário, tais como: **(i) o R é um Software Livre** (livre no sentido de liberdade) distribuído sob a Licença Pública Geral, podendo ser livremente copiado, distribuído, e instalado em diversos computadores.; **(ii) os códigos-fontes R estão disponíveis para os usuários, e atualmente são gerenciados por um grupo chamado** R Development Core Team

Fonte: Olivoto(2024)

O que é o software RStudio?

O **RStudio** é um ambiente de desenvolvimento de códigos em linguagem R. Ele funciona como uma espécie de interface, para utilizá-lo é necessário já possuir o R instalado em seu computador. **A principal vantagem de se utilizar o RStudio é que ele permite que o usuário realize suas análises de maneira muito mais organizada e intuitiva.**

Fonte: Carvalho (2024)

Fonte: Olivoto (2024)

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

[Arquivo R]

[Área de trabalho]

Figura 1.R

Source on Save | Run | Source

```
1  ################################################
2  ############### Pontos críticos ################
3  ###############    Logístico    ################
4  ################################################
5
6  require(MASS)
7  require(lmtest)
8  require(car)
9
10 dados=read.table("dados.txt",header=T)
11
12 ####### Valor inicial
13
14 plot(Ger~Dias,data=dados)
15 logi<-function(x,b0,b1,b2){
```

11:1 (Untitled) R Script

[Script]

Environment | Connections

Import Dataset | List

Global Environment

Data	
dados	32 obs. of 10 variables
log	List of 6
start	List of 3
Values	
b0	880
b1	1.5
b2	0.19
Functions	
logi	function (x, b0, b1, b2)

Console | R Markdown

C:/Users/LLenovo/Desktop/Curso R/

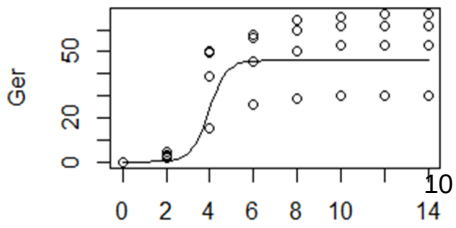
```
> summary(log)

Formula: Ger ~ b0/(1 + exp(b1 - b2 * Dias))

Parameters:
   Estimate Std. Error t value Pr(>|t|)
b0  51.3897     2.8939  17.758   <2e-16 ***
b1   6.1616     3.7323   1.651   0.1096
b2   1.8012     0.9784   1.841   0.0759 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.72 on 29 degrees of freedom

Number of iterations to convergence: 9
Achieved convergence tolerance: 4.31e-06
```

[Console]

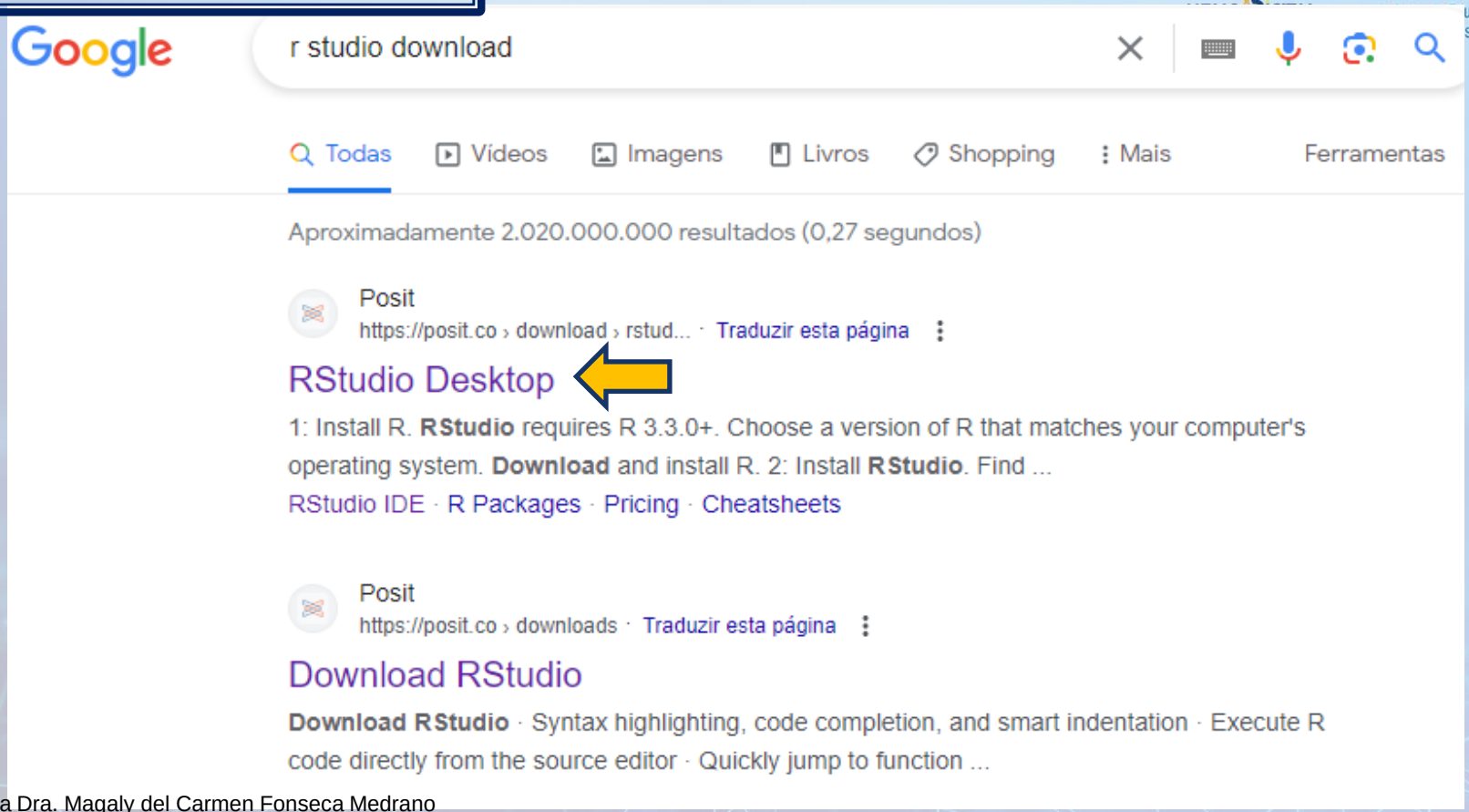
History | Files | Plots | Packages | Help | Viewer

Zoom | Export

[Output]

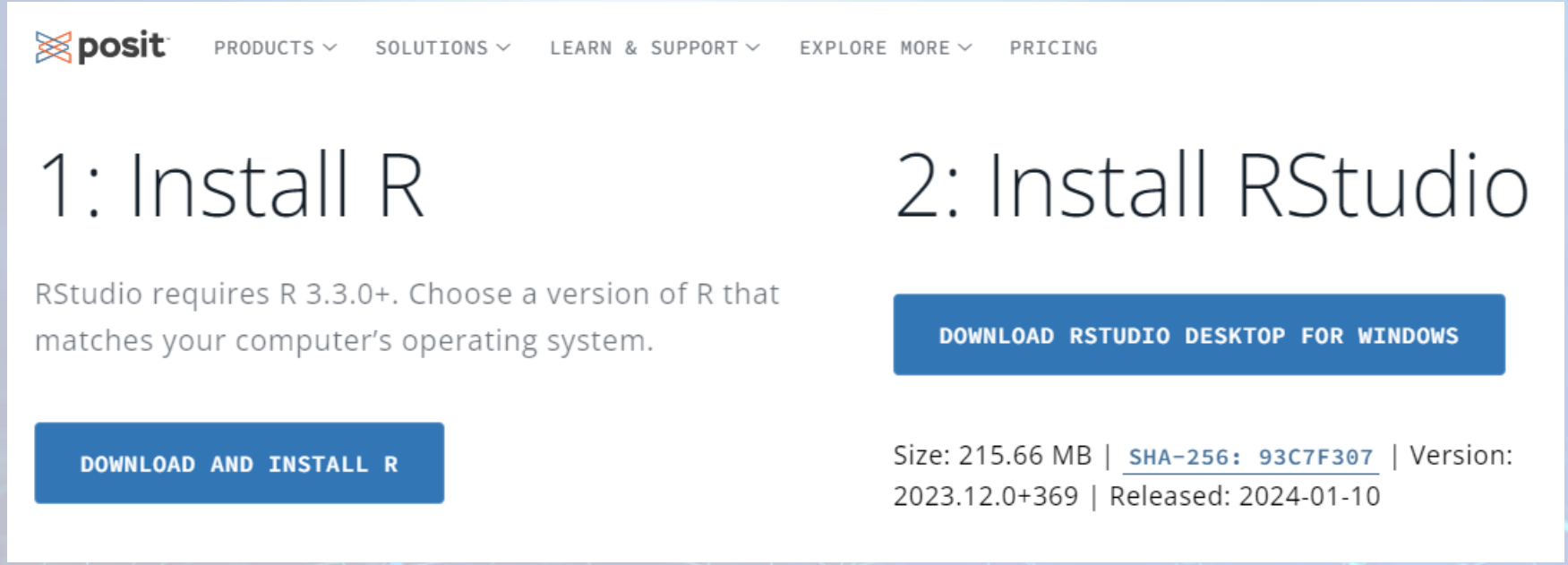


Instalação R/RStudio

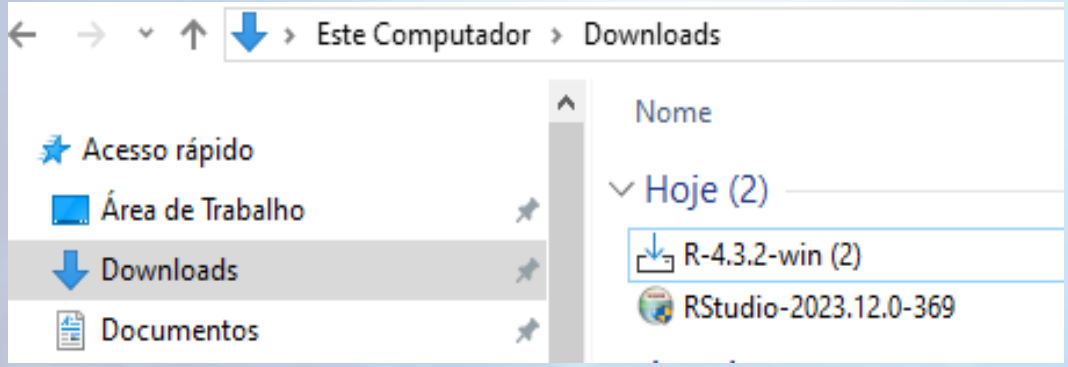


Instalação R/RStudio

https://posit.co/download/rstudio-desktop/



Instalação R/RStudio

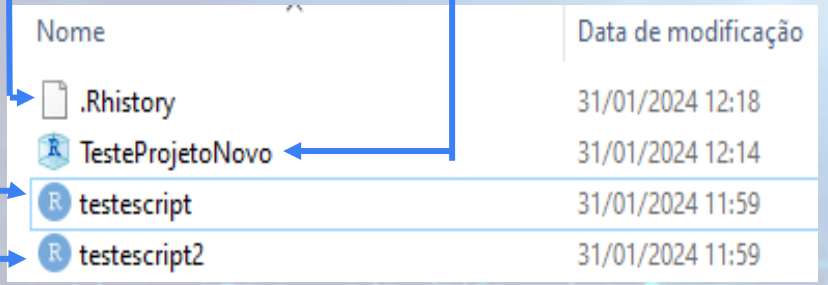


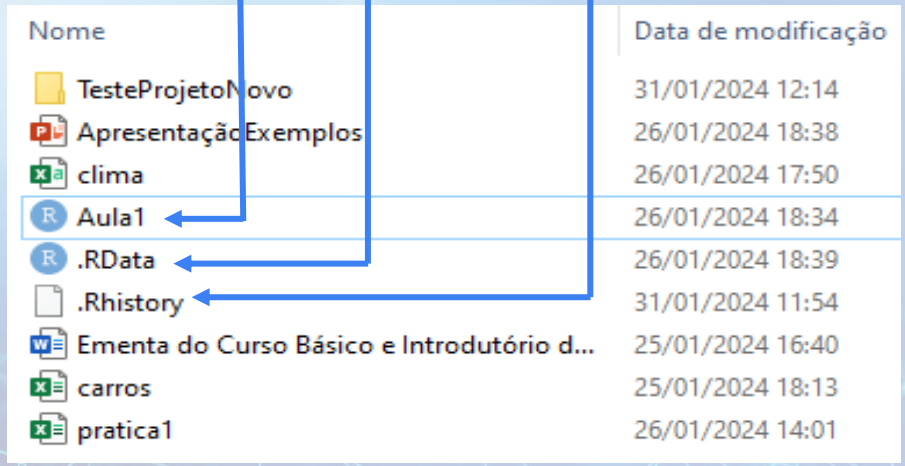
Instalar primeiro R normal e posteriormente RStudio e dar continuidade ao processo de instalação dependendo do sistema: Windows, Linux, Mac, etc.

Arquivos

Os R gera alguns tipos de arquivos que podem ser salvos. Os principais são:

- **.R** - Usado para salvar códigos criados e rotinas de análises (*scripts*)
- **.RData** - Usado para salvar os objetos da área de trabalho (*workspace*)
- **.Rhistory** - Usado para salvar o histórico dos comandos executados (normalmente salvo automaticamente).
- **.Rproj** - Guarda informações sobre projetos





1.3 Noções de Sintaxe e Estrutura de Funções

- Adicionar comentário
? - Obter ajuda de função
?? - Realizar buscar
<- ou = - Atribuir objeto (direita para esquerda)
-> - Atribuir objeto (esquerda para direita)
+ - Somar
- - Subtrair
* - Multiplicar
/ - Dividir
^ - Potencializar (direita para esquerda)
%*% - Multiplicar matrizes
< - Comparar, menor
> - Comparar, maior
<= - Comparar, menor ou igual
>= - Comparar, maior ou igual
== - Comparar, exatamente igual
!= - Comparar, diferente

& - Lógico, critério aditivo E. Operação elementar
| - Lógico, critério aditivo OU. Operação elementar
&& - Lógico, E
|| - Lógico, OU
~ - Fórmula estatística
FALSE ou F - Argumento lógico falso
TRUE ou T - Argumento lógico verdadeiro
NA - Indeterminado (Not Avaiable)
NaN - Indeterminado (Not a Number)
Inf - Infinito
NULL - Objeto nulo
c() - Concatenar valores em vetor
factor() - Criar fator
ordered() - Criar fator ordenado
data.frame() - Criar tabela de dados (data.frames)
matrix() - Criar matriz
list() - Criar lista

Fonte: (Debastiani, 2020)

1.3 Noções de Sintaxe e Estrutura de Funções

Calculadora

```
2+2 # Soma
[1] 4
8-3 # Subtração
[1] 5
3*8 # Multiplicação
[1] 24
8/2 # Divisão
[1] 4
2^8 # Potências
[1] 256
(2+4)/7 # Prioridade de solução
[1] 0.8571429
```

Ordem de parêntesis:
1-()
2-{}
3-[] é usado em vetores

Shortcuts

<Ctrl>+<Enter> = Run → **Rodar Script**

<Ctrl>+<L> → **Limpa a área do console**

Fazer ao vivo

1.3 Noções de Sintaxe e Estrutura de Funções

Funções

Função (argumento1=valor1,argumento2=valor2)

Tipos de Funções

1. Operações Aritméticas
2. Atribuição de Variáveis
3. Vetores e Indexação
4. Estatística Básica
5. Estrutura de Controle de Fluxo

1.3 Noções de Sintaxe e Estrutura de Funções

Funções de Operações Aritméticas

```
# Soma
resultado_soma <- 5 + 4
print(resultado_soma)  # Resultado: 9

# Raiz quadrada
   sqrt(9)
# Resultado: 3

# Logaritmos
log10(100)
# Resultado: 2

# Divisão
resultado_divisao <- 20 / 5
print(resultado_divisao)  # Resultado: 4
```

Funções de Atribuição de Variáveis

```
# Atribuição de variáveis
meu_numero <- 15
meu_texto <- "Olá, Mundo!"

print(meu_numero)
print(meu_texto)
```

Funções de Vetores e Indexação

```
# Criar um vetor
meu_vetor <- c(2, 4, 6, 8, 10)

# Acessar elementos do vetor
elemento_terceiro <- meu_vetor[3]
print(elemento_terceiro)  # Resultado: 6
```

1.3 Noções de Sintaxe e Estrutura de Funções

Funções de Estatística Básica

```
# Calcular a média
vetor_media <- mean(meu_vetor)
print(vetor_media)

# Calcular o desvio padrão
vetor_desvio_padrao <- sd(meu_vetor)
print(vetor_desvio_padrao)
```

Estrutura de Controle de Fluxo

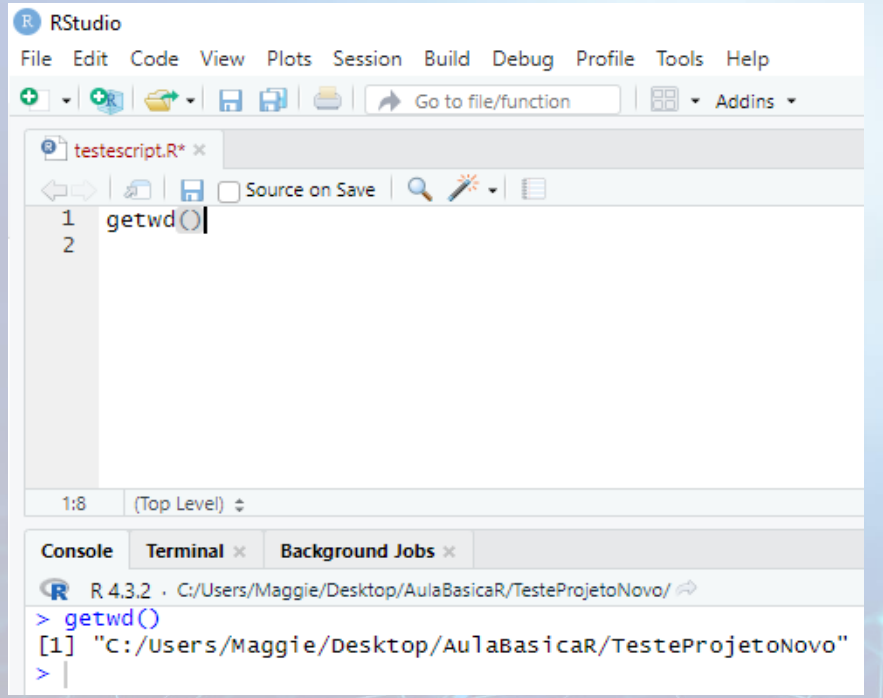
```
# Estrutura condicional (if-else)
idade <- 20

if (idade >= 18) {
  print("Você é maior de idade.")
} else {
  print("Você é menor de idade.")
}
```

Porque é importante usar ou criar funções?

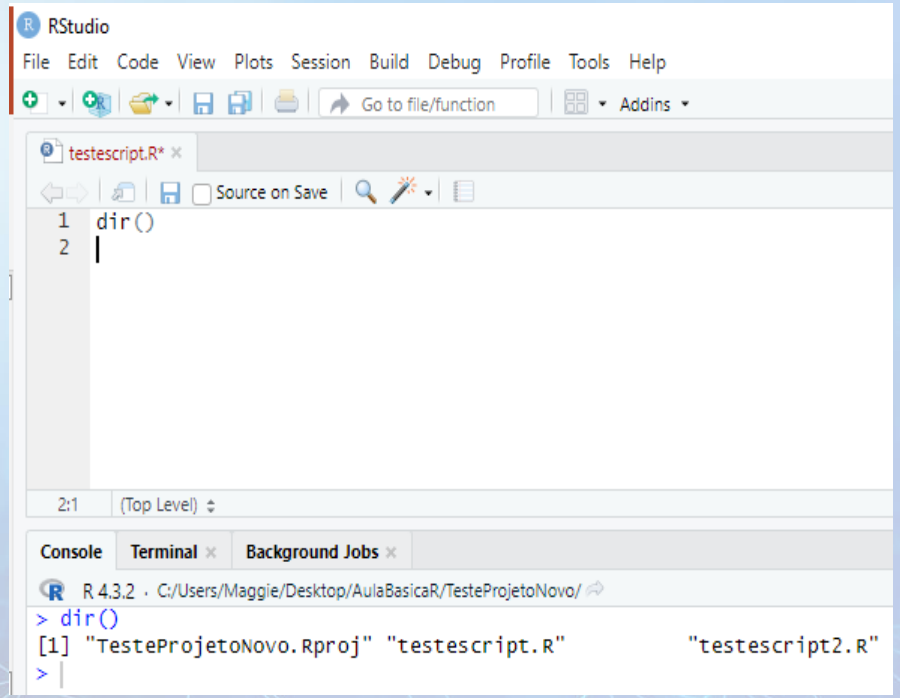
Compactar grupo de comandos e fazer o processo da análise mais rápida e eficiente

1.4 Comandos Básicos

getwd() *# Mostrar o diretório de trabalho atual*



dir() *# Listar os arquivos do diretório*



setwd() *# Mudar o diretório de trabalho*

```
1  getwd
2  setwd("C:/Users/Maggie/Desktop/AulaBasicaR")
3
```

3:1 (Top Level)

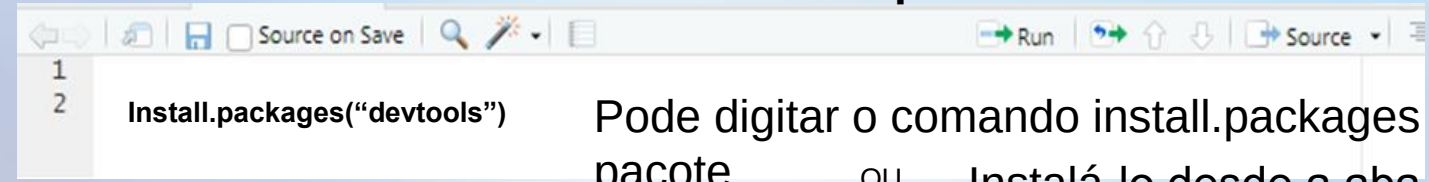
Console | Terminal | Background Jobs

R 4.3.2 · C:/Users/Maggie/Desktop/AulaBasicaR/

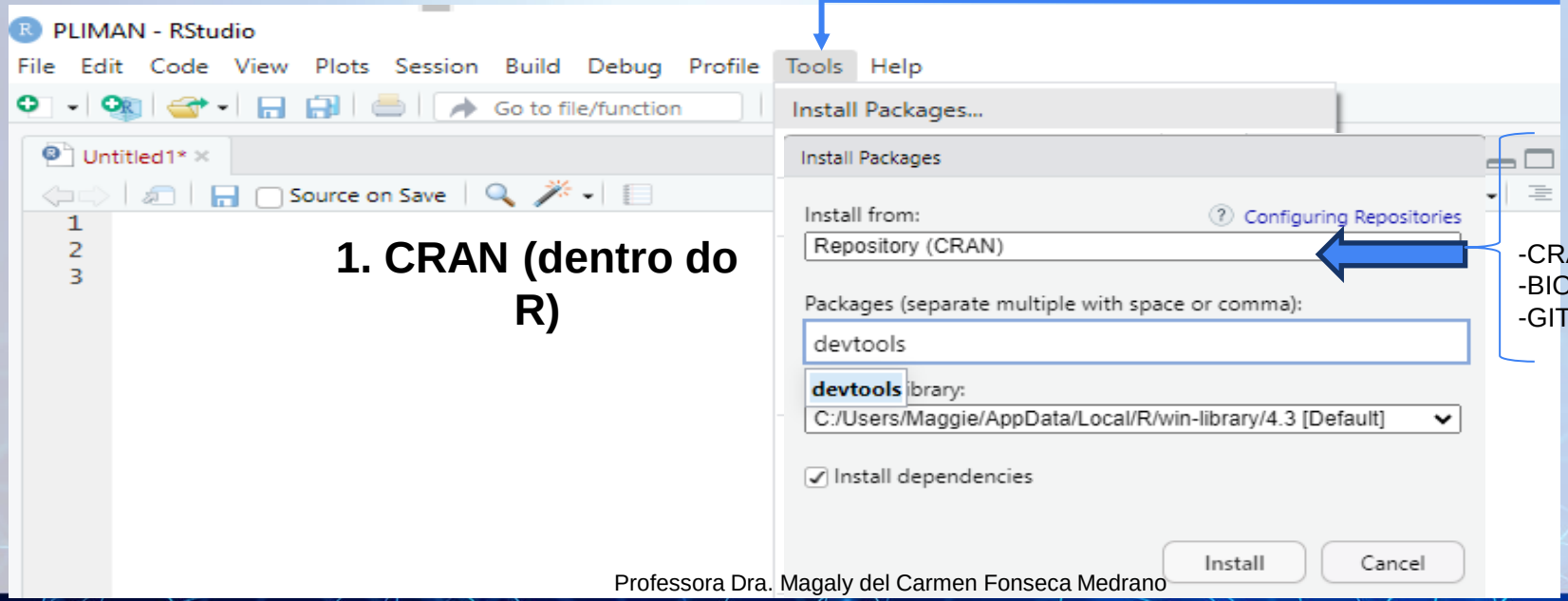
```
> dir()
[1] "TesteProjetoNovo.Rproj" "testescript.R"          "testescript2.R"
> getwd()
[1] "C:/Users/Maggie/Desktop/AulaBasicaR/TesteProjetoNovo"
> setwd("C:/Users/Maggie/Desktop/AulaBasicaR")
> C:/Users/Maggie/Desktop/AulaBasicaR/TesteProjetoNovo
```

1.5 Instalação de Pacotes

Os pacotes do R podem ser extraídos de vários repositórios:



Pode digitar o comando install.packages + o nome do pacote ou Instalá-lo desde a aba "Tools"



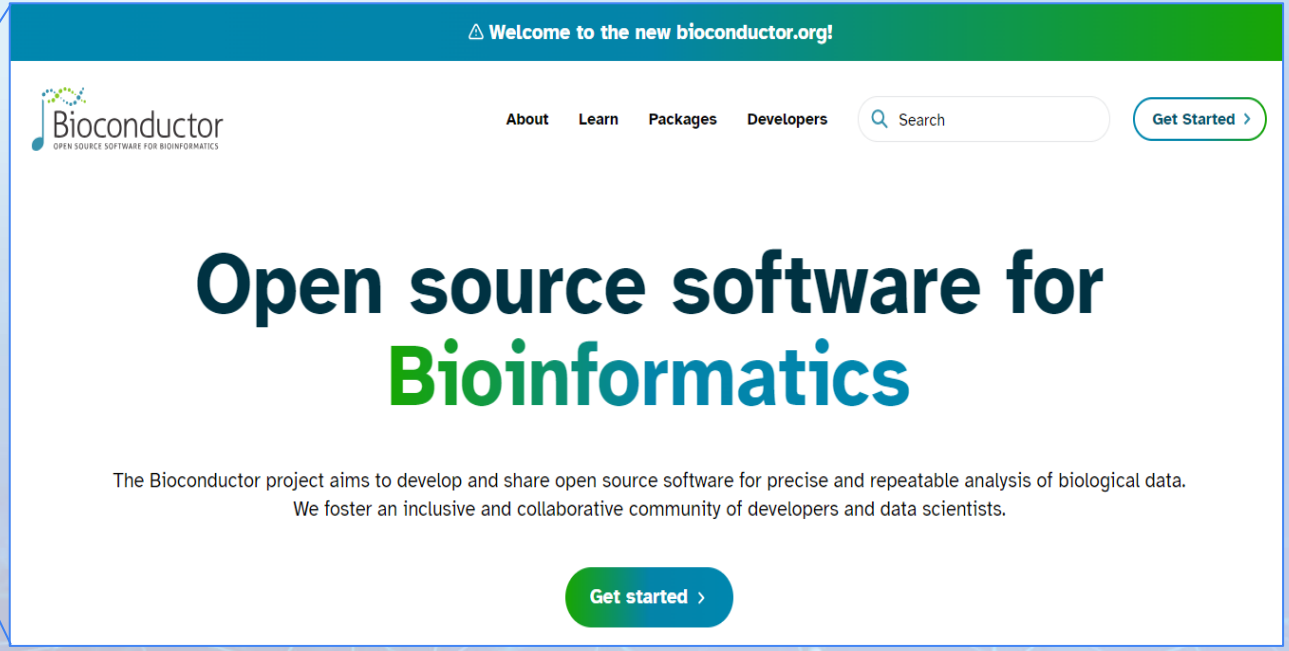
1. CRAN (dentro do R)

-CRAN
-BIOCONDUCTOR
-GITHUB

1.5 Instalação de Pacotes

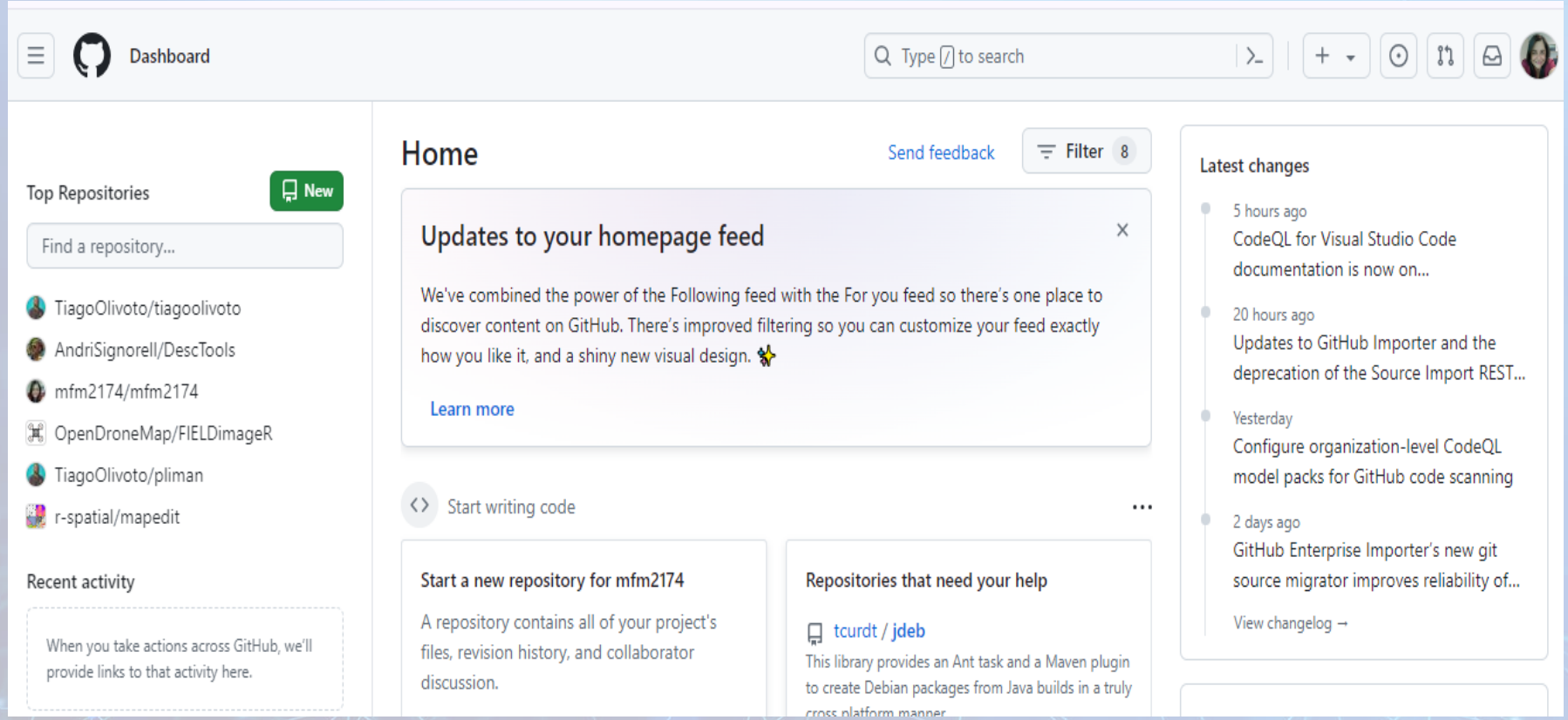
2. BIOCONDUCTOR

https://www.bioconductor.org/



3. GITHUB

https://github.com/



2. Estrutura e Manipulação de Dados (30min)

2.1 Tipos de Estrutura (Vetores, Data frames, Matrizes e Listas)

2.2 Importação e Exportação de Dados

2.1 Tipos de Estruturas

Vetores (vector) → São estruturas de dados fundamentais que **armazenam elementos do mesmo tipo.** Eles podem ser **numéricos, caracteres, lógicos** e **data**

```
numeros <- c(5,4,3,2,1)
nomes <- c("Alice", "Bob", "Charlie")
logico <- c(TRUE, FALSE, TRUE)
datas <- as.Date(c("2022-01-01", "2022-02-01")
```

Fatores → É um vetor que representa **dados categóricos**

```
genero <- factor(c("Masculino", "Feminino", "Masculino", "Feminino"))

avaliacao <- factor(c("Ruim", "Regular", "Bom", "Excelente"))
```

Indexar Vetores

```
numeros[4]
[1] 2
```

Acessa o 4to valor

```
numeros[c(1,3,5)]
[1] 5,3,1
```

Acessa o 1er, 3ero e 5to valor

Fonte: Carvalho(2024)

Data frames (Tabelas)

Um data frame é uma **estrutura bidimensional que organiza dados em linhas e colunas**, semelhante a uma planilha. **Cada coluna pode conter diferentes tipos de dados**, mas todas as colunas devem ter o mesmo número de linhas

```
nomes<-
c("ARTUR","ROBERTO","HELCI","NATALIA","LANA","ROGERIO","E
LAINE","JAIRO","LETICIA")

genero<-c("M", "M", "M", "F", "F", "M", "F", "M", "F")

faixaecon<-c("A", "B", "C", "A ", "B", "C", "D", "A", "D")

avaliacao<-c("excelente", "bom", "bom", "ruim", "pessimo", "regular",
"regular", "mbom", "bom")

filhos<-c("TRUE", "FALSE", "TRUE", "FALSE", "FALSE", "FALSE",
"TRUE", "TRUE", "TRUE")


meu.df<-data.frame(nomes,genero,faixaecon,avaliacao,filhos)
```

NOME	GENERO	FAIXAECON	AVALIACAO	FILHOS
ARTUR	M	A	EXCELENTE	TRUE
ROBERTO	M	B	BOM	FALSE
HELCI	M	C	BOM	TRUE
NATALIA	F	A	RUIM	FALSE
LANA	F	B	PESSIMO	FALSE
ROGERIO	M	C	REGULAR	FALSE
ELAINE	F	D	REGULAR	TRUE
JAIRO	M	A	MBOM	TRUE
LETICIA	F	D	BOM	TRUE

Fonte: Carvalho(2024)

Matrizes

Uma matriz assim como o *data frame* é uma **estrutura bidimensional que organiza dados em linhas e colunas. Todos os elementos contidos devem ser do mesmo tipo**

n

m

Modelagem feminina				
Tamanho / Estampa	lisa	flores	dragão	pirata
P	10	5	0	16
M	15	12	10	14
G	12	7	16	11
Modelagem masculina				
Tamanho / Estampa	lisa	flores	dragão	pirata
P	8	4	8	12
M	7	13	10	20
G	15	8	16	2

$$A_1 = \begin{pmatrix} 10 & 5 & 0 & 16 \\ 15 & 12 & 10 & 14 \\ 12 & 7 & 16 & 11 \end{pmatrix}$$

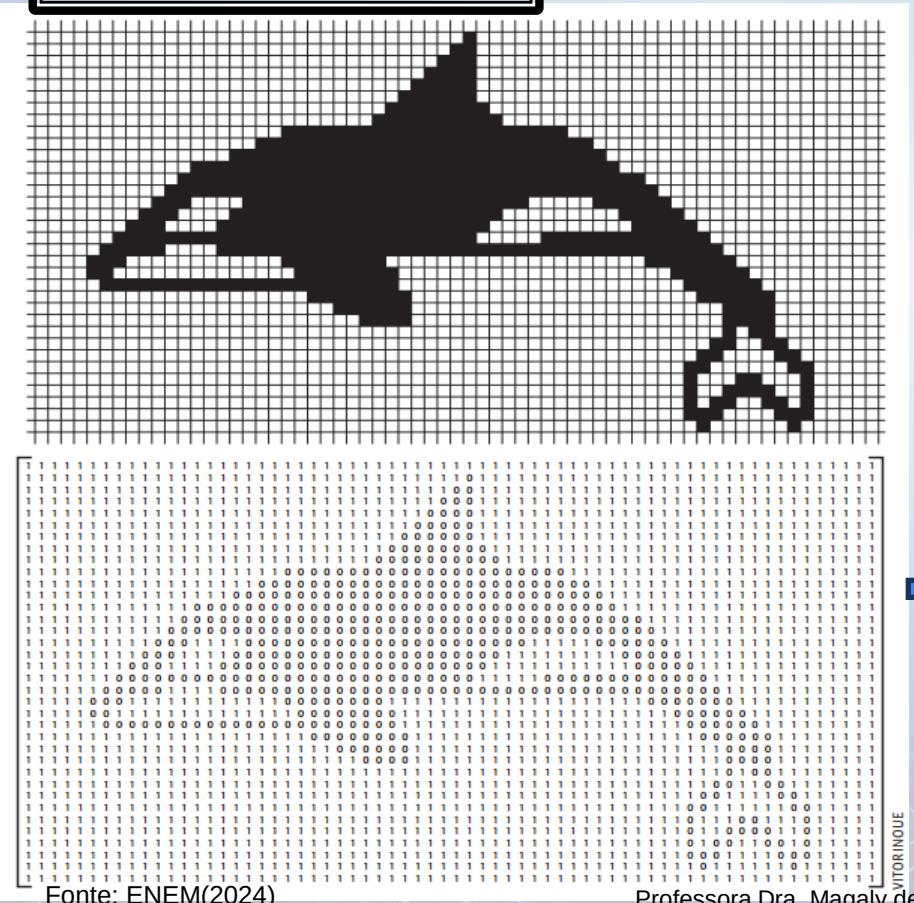
A1, $a_{12=}5$

Fila1, Coluna2

$$A_2 = \begin{pmatrix} 8 & 4 & 8 & 12 \\ 7 & 13 & 10 & 20 \\ 15 & 8 & 16 & 2 \end{pmatrix}$$

Fonte: Carvalho(2024); ENEM(2024)

Matrizes



Fonte: ENEM(2024)

A figura desenhada pelo código binário, utilizado em informática, é uma matriz

Matrizes

```
matriz_numerica <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3, byrow = TRUE)
print(matriz_numerica)
```

Matriz 2x3

```
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
```

```
6  head(matriz_numerica, n=1)
7
```

5:1 (Top Level)

Console | Terminal | Background Jobs

R 4.3.2 · C:/Users/Maggie/Desktop/AulaBasicaR/

```
> head(matriz_numerica, n=1)
     [,1] [,2] [,3]
[1,]    1    2    3
```

Mostra os dados da fila 1

```
8  tail(matriz_numerica,n=1)
9
10
```

9:1 (Top Level)

Console | Terminal | Background Jobs

R 4.3.2 · C:/Users/Maggie/Desktop/AulaBasicaR/

```
> tail(matriz_numerica,n=1)
     [,1] [,2] [,3]
[2,]    4    5    6
```

Mostra os dados da última fila

Matrizes

```
4  str(matriz_numerica)
5
```

```
> str(matriz_numerica)
 num [1:2, 1:3] 1 4 2 5 3 6
>
```

Mostra o resumo da matriz
1:2 -> 2 filas
1:3-> 3 colunas

```
7  nrow(matriz_numerica)
8
9
```

```
> nrow(matriz_numerica)
[1] 2
>
```

Mostra o # de filas

```
7  ncol(matriz_numerica)
8
9
```

```
> ncol(matriz_numerica)
[1] 3
```

Mostra o # de colunas

Matrizes

Indexar Matrizes

matriz_numerica[1,1]
[1] 1

→ Mostra o conteúdo da fila1,coluna1

```
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
```

matriz_numerica[-2, , drop = FALSE]

→ Se está excluindo a segunda linha da matriz

```
     [,1] [,2] [,3]
[,1]    1    2    3
```

drop = FALSE: é usado para garantir que o resultado seja uma matriz (e não um vetor) mesmo que tenhamos apenas uma linha após a exclusão da segunda linha

matriz_numerica <- matriz_numerica[,c(2,1,3)]

→ Está alterando a ordem das primeiras 2 colunas

```
> print(matriz_numerica)
     [,1] [,2] [,3]
[1,]    2    1    3
[2,]    5    4    6
```

Listas

Podem conter **elementos de diferentes tipos e ser multidimensionais**

```
# Criando um vetor
vetor_numerico <- c(1, 2, 3, 4, 5)

 # Criando uma matriz
 matriz_numerica <- matrix(1:6, nrow = 2, ncol = 3)

# Criando um data frame

dataframe_exemplo <- data.frame(
Nome = c("Alice", "Bob", "Charlie"),
Idade = c(25, 30, 22),
Salario = c(50000, 60000, 45000)
 )
```

```
# Criando uma lista que contém o vetor, a matriz, e o data frame

minha_lista <- list(
Vetor = vetor_numerico,
Matriz = matriz_numerica,
DataFrame = dataframe_exemplo,
OutroElemento = "Texto de exemplo",
 Logico = TRUE )

# Exibindo a lista
 print(minha_lista)
```

Listas

Indexar Listas

$ ➡ É usado para acessar elementos de uma lista ou de um data frame no R

```
30  minha_lista$Matriz
31
32
31:1  (Top Level)
Console  Terminal  Background Jobs
R 4.3.2 · C:/Users/Maggie/Desktop/AulaBasicaR/
> minha_lista$Matriz
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
30  minha_lista$DataFrame
31
32
31:1  (Top Level)
Console  Terminal  Background Jobs
R 4.3.2 · C:/Users/Maggie/Desktop/AulaBasicaR/
> minha_lista$DataFrame
     Nome Idade Salario
1   Alice    25   50000
2     Bob    30   60000
3 Charlie    22   45000
```

minha_lista[2] ➡ Mostra o conteúdo da Matriz (Sublista)

minha_lista[[2]] ➡ Dá acesso ao conteúdo da Matriz

```
print(sum(minha_lista[[2]]))   #soma todos os elementos da Matriz=21
```

minha_lista[3]

minha_lista[[3]]

2.2 Importação e Exportação de Dados

Importação

Existem diversas maneira para abrir um arquivo que contém dados:

Arquivos extensão csv

dados<-read.csv(“nomearquivo.csv”, header=TRUE, ...)
dados<-read.csv(file.choose())
dados<-read.csv(“C:/Users/Maggie/Desktop/AulaR/nomedoarquivo.csv”’)

Arquivos extensão xls

Precisa-se instalar o pacote readxl
dados<-read_excel(caminho_do_arquivo) ou
dados<-read_excel(caminho_do_arquivo, sheet=2)

dados<-read_excel(file.choose())
dados<-read_excel("C:/Users/Maggie/Desktop/AulaBasicaR/nomedoarquivo.xlsx")

2.2 Importação e Exportação de Dados

Importação

```
############  LER DADOS CSV
library(readr)

clima <- read.csv("clima.csv", header = TRUE)
clima=read.csv(file.choose())
clima=read.csv("C:/Users/Maggie/Desktop/AulaBasicaR/clima.csv")

############  LER DADOS DE EXCEL

library(readxl)

carros<-read_excel("C:/Users/Maggie/Desktop/AulaBasicaR/carros.xlsx")
Carros<-read_excel(file.choose())
carros<-read_excel("C:/Users/Maggie/Desktop/AulaBasicaR/nomedoarquivo.xlsx")
```

Fazer ao vivo

2.2 Importação e Exportação de Dados

Exportação

```
library(readxl)

carros<-read_excel("C:/Users/Maggie/Desktop/AulaBasicaR/carros.xlsx")
```

1) Função Export

```
library(rio)
```

```
export(carros, file="C:/Users/Maggie/Desktop/AulaBasicaR/carrosteste1.txt")
```

2) Função Write.Table / Write.csv (data frames)

```
write.table(carros,"C:/Users/Maggie/Desktop/AulaBasicaR/carrosteste2.txt" )
```

```
write.csv(carros,"C:/Users/Maggie/Desktop/AulaBasicaR/carrosteste3.csv")
```

3. Estatística Descritiva (30min)

3.1 Medidas de Tendência Central (Média, Mediana, Moda)

3.2 Medidas de Variabilidade (Variância, Desv.Padrão, Coef. Variação)

3.3 Medidas de Posições (Percentis e Quartis)

3.4 Tipos de Gráficos

Tipos de Estatística

ESTATÍSTICA

DESCRITIVA

Descreve e resume um conjunto de dados

INFERENCIAL

Faz inferências sobre uma população com base em amostras tiradas dessa população

- **NÃO PARAMÉTRICA** → Aplica métodos usando os dados **não** seguem uma distribuição específica (normal) ou quando a suposição sobre a distribuição é desconhecida.
- **PROBABILÍSTICA** → Trata do estudo de fenômenos aleatórios e incertos
- **MULTIVARIADA** → Analisa simultaneamente múltiplas variáveis
- **COMPUTACIONAL** → Usa técnicas computacionais avançadas, como simulação Monte Carlo, métodos de bootstrap e algoritmos de aprendizado de máquina
- **BAYESIANA** → Avalia hipóteses pela máxima verossimilhança, uma decorrência imediata da fórmula de Bayes

Fonte: Aguado(2023);Beneti(2011); Castañeda(2016); Hair et al.(2009); Kopczewska(2020); Piegorsch et al.(2022); Silvestre, A.(n.d.)

Análises aplicáveis na estatística descritiva:

3.1 Medidas de centralidade: Média, mediana, moda

3.2 Medidas de variabilidade: Variância, desvio padrão, coeficiente de variação

3.3 Medidas de posições: quartis e percentis

Fonte: Asth(2024);Beneti. (2011);

3.1 Medidas de Centralidade: Média, Mediana e Moda

```r
clima <- read.csv("clima.csv", header = TRUE)

# Calcular a média
media <- mean(clima$tmax)

# Calcular a mediana
mediana <- median(clima$tmax)
```

A **média (M_e)** é calculada somando-se todos os valores de um conjunto de dados e dividindo-se pelo número de elementos deste conjunto.

A **Mediana (M_d)** representa o **valor central** de um conjunto de dados

Fazer ao vivo

3.1 Medidas de Centralidade: Média, Mediana e Moda

A **Moda** (M_o) representa o **valor mais frequente** de um conjunto de dados

1)

```
## criando uma série de dados qualquer

w <- c(1, 2, 3, 4, 4, 4, 5, 6, 7)

## visualmente encontrando a moda da série

table (w)
```

```
## w
## 1 2 3 4 5 6 7
## 1 1 1 3 1 1 1
```

2)

```
# Calcular a moda
#1era opção - menos elaborada
moda<-table(clima$tmax)
print(moda)

#2da opção - mais elaborada
moda <- names(sort(-table(clima$tmax)))[1]
print(moda)
```

Arquivo clima.csv

3.2 Medidas de Variabilidade: Variância, Desvio Padrão e Coeficiente de Variação

O que variância? é uma medida de dispersão que indica o quão distantes os valores estão da média

Mais difícil de interpretar diretamente, pois está em unidades quadradas

Variância Populacional

$$\sigma^2 = \frac{\sum_{i=1}^{n} (X_i - \overline{X})^2}{n}$$

Onde,

σ^2: variância

x_i: valor analisado

$\overline{X}$: média aritmética do conjunto

n: número de dados do conjunto

Variância Amostral

$$s^2 = \frac{\sum_{i=1}^{n} (X_i - \overline{X})^2}{n - 1}$$

s^2: variância

Desvio Padrão

$$DP = \sqrt{\sigma^2} \text{ ou } DP = \sqrt{s^2}$$

É a raiz quadrada da variância e fornece uma medida de dispersão em torno da média

Reis;Reis(2002)

3.2 Medidas de Variabilidade: Variância, Desvio Padrão e Coeficiente de Variação

Exemplo:

Se a altura de um grupo de pessoas tem uma **variância** de 64 cm quadrados ($\sigma^2=64cm^2$), significa que em média, os dados estão a 8cm ($\sigma=8cm$)(**Desvio Padrão**) da média.

Coeficiente de Variação Indica a variabilidade de dispersão dos dados com respeito à média em %

```
# Calcular a variância
variancia <- var(clima$tmax)
```

```
3.54792634615385
```

```
# Calcular o desvio padrão
desvio_padrao <- sd(clima$tmax)
```

```
1.88359399716442
```

```
# Calcular o coeficiente de variação
coef_variacao <- desvio_padrao / mean(clima$tmax) * 100
```

```
7.1374045293572
```

Interpretação do Coef.Variação: o desvio padrão dos valores de temperatura máxima é cerca de 7.13%

3.2 Medidas de Variabilidade: Variância, Desvio Padrão e Coeficiente de Variação

```
######  RESUMIR DADOS COM GROUP_BY e SUMMARISE ###########
# Serve para observar a dispersão dos pontos em relação
# à média ou a uma linha de tendência

library(readr)
library(dplyr)     #filtragem, agrupação, agregação
library(lubridate) #lida com datas e tempos
library(ggplot2)   #gráficos
```

```
dado<-read_csv("clima.csv")%>%group_by()
dado<-read_csv("clima.csv")%>%group_by(month(data)) #cria coluna do mês (só o número)
dado<-read_csv("clima.csv")%>%group_by(mes=month(data)) #nomea coluna como: mes
dado<-read_csv("clima.csv")%>%group_by(mes=month(data,label=TRUE)) #coloca nome do mês no
                                                                   #lugar de números


dado<-read_csv("clima.csv")%>%
  group_by(mes=month(data,label=TRUE))%>%
  summarise(media=mean(tmax))

dado<-read_csv("clima.csv")%>%
  group_by(mes=month(data,label=TRUE))%>%
  summarise(media=mean(tmax),desv.pad=sd(tmax))
```

3.2 Medidas de Variabilidade: Variância, Desvio Padrão e Coeficiente de Variação

```
###### Gráfico de Dispersão/Histograma para ver variação com respeito à Média

#Média: 26.39

# Gráfico de Dispersão (Deve-se Especificar eixo x,y)
ggplot(dado, aes(x=media,y=desv.pad))+geom_point()

# Gráfico de Histograma
ggplot(dado, aes(x=media,y=desv.pad))+geom_col() #Histograma
```

3.3 Medidas de Posições

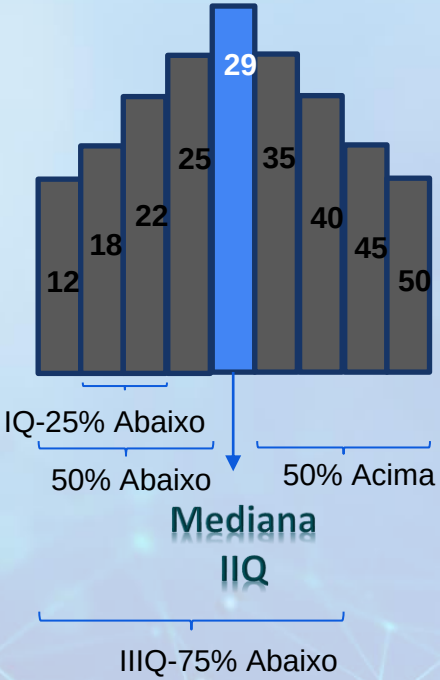
Quartis e Percentis

Um quantil é um conceito estatístico que divide um conjunto de dados ordenados em partes iguais ou em porcentagens específicas. Descrevem a posição relativa de um valor dentro desse conjunto de dados.

Conjunto de Dados:

12,18,22,25,28,30,35,40,45,50

(28+30)/2=29 → Mediana



Quartis:

Primeiro Quartil (Q1): É o valor que deixa 25% dos dados abaixo e 75% acima. Também é conhecido como o 25º percentil.
Segundo Quartil (Q2): É a mediana, representando o valor que divide os dados ao meio (50% abaixo e 50% acima). Também é conhecido como o 50º percentil.
Terceiro Quartil (Q3): É o valor que deixa 75% dos dados abaixo e 25% acima. Também é conhecido como o 75º percentil.

3.3 Medidas de Posições

Quartis e Percentis

```
# Quantis & Percentis
dados <- c(12, 18, 22, 25, 28, 30, 35, 40, 45, 50)

# Calcular os quantis teóricos
quantis_teóricos <- quantile(dados, probs = seq(0.25, 0.75, 0.01)) #25%, 75%, 0.01=compasso

#Determinar a distribuição dos quantis
quantile(dados)
```

3.3 Medidas de Posições

Quartis e Percentis

```
###### GRÁFICO DE QUANTIS & PERCENTIS

qqplot(quantis_teóricos, dados,
       xlab = "Quantis teóricos",
       ylab = "Dados",
       main = "Gráfico Q-Q de quantis e percentis")

abline(0, 1, col = "red")  # Adiciona linha de referência no eixo Y=0,1=45 graus

####### GRÁFICO Determinar a normalidade dos quantis

qqnorm(dados) #Quantidades Teoricas oscila entre -3 a 3 (Distribuição Normal)
qqline(dados) #Linha de referência da distribuição normal
```

Também conhecido como Q-Q plot, é uma ferramenta gráfica útil para comparar duas distribuições de probabilidade

3.4 Medidas de Centralidade-GRÁFICOS

HISTOGRAMA

Visualiza a distribuição de dados quantitativos, forma de distribuição, identifica outliers, etc.

```
hist(clima$tmax)

# Modificar info eixo x e y
hist(clima$tmax, main = "Histograma", xlab = "Valores", ylab = "Frequência",
     col = "lightblue", border = "black")

# Adicionar legenda
legend("topright", legend = c("Percentis & Quantis"), fill = "lightblue", border = "black")
```

3.4 Medidas de Centralidade-GRÁFICOS

BARPLOT

Visualiza dados categóricos, Compara frequências, identifica padrões e tendências

Dados que representam categorias (gênero, país) ou grupos discretos (absolutos)

```
library(readxl)

salarios<-read_excel('salarios.xlsx',3)

barplot(salarios$RENDA_MENSAL)

#Adicionando títulos e cor
barplot(salarios$RENDA_MENSAL, names.arg = salarios$NIVEL, col = "skyblue",
     xlab = "Categorias", ylab = "Salários R$", main = "Gráfico de Barras")

# Adicionar legenda
legend("topright", # Posição da legenda
       legend = c("A:Rico", "B:Classe Média", "C: Pobre"),
       cex = 0.8) # Rótulos da legenda
```

3.4 Medidas de Centralidade-GRÁFICOS

PIE/PIZZA

Visualiza dados categóricos

```
salarios<-read_excel('salarios.xlsx',4)
pie(salarios$QUANTIDADE)

# Vetor de cores
cores <- c("skyblue", "orange", "purple")

dado <- salarios$QUANTIDADE # Suponha que esses sejam os dados para "Rico",
                            # "Classe Média" e "Pobre"

# Criar gráfico de pizza
pie(dado, labels = c("55%", "15%", "30%"), col = cores,
    main = "Gráfico de Pizza", radius = 1)

# Adicionar legenda
legend("topright", # Posição da legenda
       legend = c("Pobre", "Rico", "Classe Média"),
       fill = cores,
       cex = 0.6)
```

3.4 Medidas de Centralidade-GRÁFICOS

BOXPLOT

Visualizar a distribuição de dados numéricos, identificar outliers, avaliar assimetria, comparar distribuições

```
salarios<-read_excel('salarios.xlsx',3)

genero<- salarios$GENERO

# Converter a variável GENERO em um fator
salarios$GENERO <- factor(salarios$GENERO)

#Criar Boxplot

boxplot(RENDA_MENSAL ~ GENERO, data = salarios, xlab = "Gênero", ylab = "Salário R$",
        main = "Distribuição dos Salários por Gênero")
```

Referências

Aguado, S. C. (2023). *Estatística e métodos quantitativos aplicados a finanças*. Editora Senac São Paulo.

Asth, R. (2024). *Variância e desvio padrão: O que são, fórmulas, como calcular e exercícios*. Toda Matéria. https://www.todamateria.com.br/variancia-e-desvio-padrao/

Beneti, M. (2011). *Estatística Básica*. Clube de Autores.

Carvalho, C. de. (2024). *Introdução ao R*. https://www.est.ufmg.br/~cristianocs/Pacotes2021/Intro.html#3

Castañeda, D. F. N. (2016). *Estatística Não Paramétrica*. Clube de Autores.

Chein, F. (2019). *Introdução aos modelos de regressão linear: Um passo inicial para compreensão da econometria como uma ferramenta de avaliação de políticas públicas /*. Enap.

Debastiani, V. J. (2020). *Introdução ao R*. https://vanderleidebastiani.github.io/tutoriais/Introducao_ao_R.html#no%C3%A7%C3%B5es_b%C3%A1sicas

ENEM, 2024. ([s.d.]). *Matrizes: Prateleiras matemáticas | Curso Enem Play | Guia do Estudante*. Recuperado 19 de abril de 2024, de https://guiadoestudante.abril.com.br/curso-enem/matrizes-prateleiras-matematicas

Georgiev, G. (2020, dezembro 9). *The p value – definition and interpretation of p-values in statistics*. GIGAcalculator Articles. https://www.gigacalculator.com/articles/p-value-definition-and-interpretation-in-statistics/

Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. E., & Tatham, R. L. (2009). *Análise multivariada de dados—6ed*. Bookman Editora.

HCODE. (2024). *Qual a diferença entre linguagem compilada, interpretada e JIT (Just in Time)?* Hcode. https://hcode.com.br/blog/qual-a-diferenca-entre-linguagem-compilada-interpretada-e-jit-just-in-time

Kopczewska, K. (2020). *Applied Spatial Statistics and Econometrics: Data Analysis in R*. Routledge.

Olivoto, T. (2024). *Capítulo 1 Introdução ao ambiente R | Software R para avaliação de dados experimentais*. https://tiagoolivoto.github.io/e-bookr/intro.html

Piegorsch, W. W., Levine, R. A., Zhang, H. H., & Lee, T. C. M. (2022). *Computational Statistics in Data Science*. John Wiley & Sons.

Reis, E. A., & Reis, I. A. (2002). *Análise Descritiva de Dados. Relatório Técnico do Departamento de Estatística da UFMG*. www.est.ufmg.br

Rosa, D. (2021, dezembro 20). *Linguagens de programação interpretadas x compiladas: Qual é a diferença?* freeCodeCamp.org. https://www.freecodecamp.org/portuguese/news/linguagens-de-programacao-interpretadas-x-compiladas-qual-e-a-diferenca/

Santos, M. (Diretor). (2024). *Introdução a Linguagem R | R para iniciantes | O que de fato é a Linguagem R e como aprender?* https://www.youtube.com/watch?v=jTMOuafD_4M

Silvestre, A. ([s.d.]). *Analise de Dados e Estatistica Descritiva*. Escolar Editora.

UFRGS (Diretor). (2018, agosto 14). *Erros Tipo 1 (Alfa) e Tipo 2 (Beta)—Monitoria de Epidemiologia FAMED*. https://www.youtube.com/watch?app=desktop&v=5jyi2kiIQJ0

Para citações do trabalho:

Fonseca-Medrano, M. (2024, maio). Estatística Descritiva no Software R – Ciência de Dados. *I Seminário de Extensão do Programa Digital da UFMS*, on-line. Cidade Universitária, MS: Graduação em Ciência de Dados.

Obrigada pela Participação!

Dra. Magaly del Carmen Fonseca Medrano

E-mail: magaly.fonseca@ufms.br
fonsecatolke@gmail.com