Faculdade Multiversa

INTRODUÇÃO A ANÁLISE DE DADOS COM A LINGUAGEM R

Conceitos e exemplos práticos

A ANÁLISE DE DADOS:

A análise de dados é a arte de transformar dados em conhecimentos e insights relevantes. Ou seja, comparar e agregar as informações brutas para entender o que os dados nos dizem.

A análise de dados pode ser dividida em quatro níveis:

1. **Análise descritiva**: Consiste na descrição das principais características de um conjunto de dados, listando e resumindo valores, certas vezes de apenas uma variável.
2. **Análise exploratória**: agora, além de toda etapa descritiva, a análise abarca também a correlação entre variáveis, usando técnicas como regressões e análise de variância.
3. **Análise preditiva**: utiliza uma série histórica dos dados a fim de realizar previsões sobre eventos futuros.
4. **Análise prescritiva**: neste nível, a partir do acúmulo das análises anteriores, o objetivo é gerar a tomada de ações ou sugestões, de forma automática ou semiautomática.

Faculdade Multiversa

CONHECENDO A ORIGEM DA LINGUAGEM R

R é um conjunto integrado de recursos de software para manipulação de dados, cálculo e exibição de gráficos. Foi desenvolvido a partir da linguagem S (que também é usada numa versão comercial – o S-Plus), que tem suas origens nos laboratórios da AT&T no final dos anos 80. Em 1995 dois professores de estatística da Universidade de Auckland, na Nova Zelândia, iniciaram o “Projeto R”, com o intuito de desenvolver um programa estatístico poderoso baseado em S, e de domínio público.

VANTAGENS E DESVANTAGENS

Vantagens

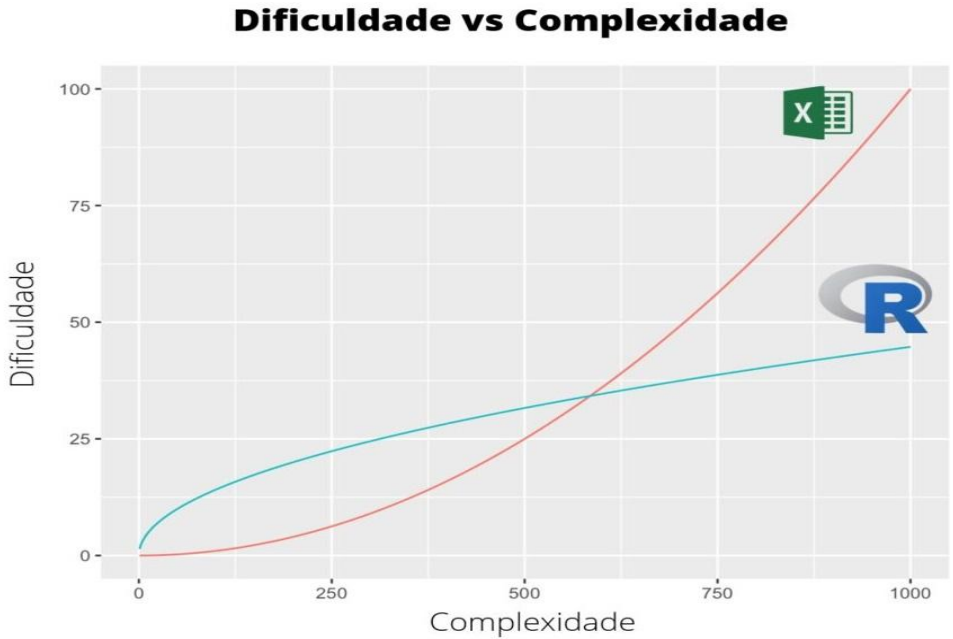
- Rápido, flexível e gratuito (Licença GNU)
- Pesquisadores de Estatística fornecem os seus métodos em pacotes de R
- Grande diversidade de gráficos, com inúmeras possibilidades de customização.
- Comunidade de usuários ativos
- Excelente para executar simulações, web scraping, machine learning, processos de ETL…
- Interfaces com software de armazenamento de banco de dados (SQL)

Desvantagens

- Curva de aprendizagem significativa*
- Não possui interface gráfica.
- Não há suporte comercial
- O processamento de grandes conjuntos de dados é limitado pela RAM

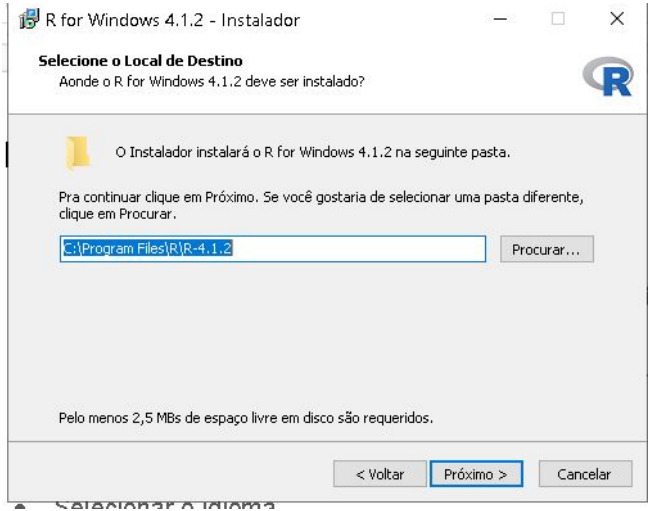
Faculdade Multiversa

Comparando R e Excel



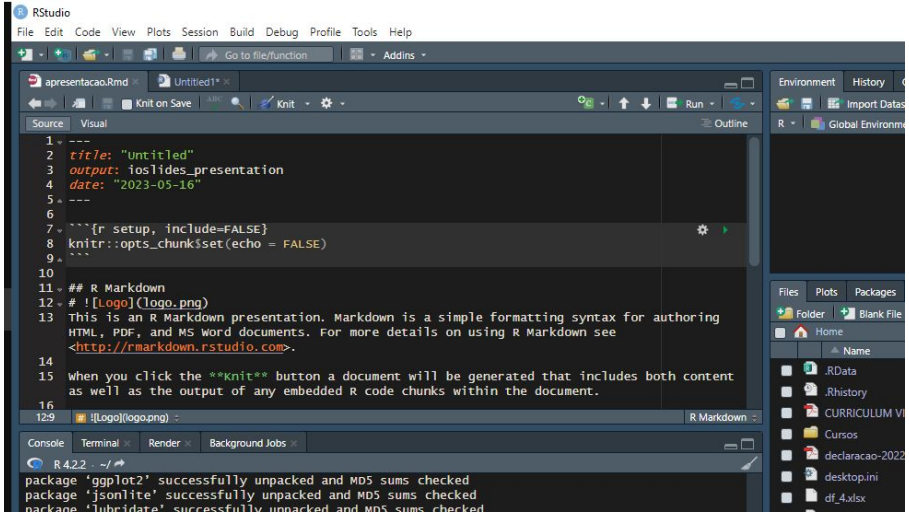
Instalação do R

- Obter o arquivo executável em:
- Acessar https://cran.r-project.org/mirrors.html
- Escolher o espelho CRAN de sua preferência (padrão: https://cloud.r-project.org/)
- Selecionar o sistema operacional utilizado, no caso Windows, clicar em **base** e em seguida **Download R... for Windows** nas páginas seguintes
- Executar o arquivo baixado
- Selecionar o idioma
- Clicar em **Próximo** nas telas seguintes



R e Rstudio

O Rstudio é uma IDE para R e Python, com um console, editor de realce de sintaxe que suporta a execução direta de código e ferramentas para plotagem, histórico, depuração e gerenciamento de espaço de trabalho.



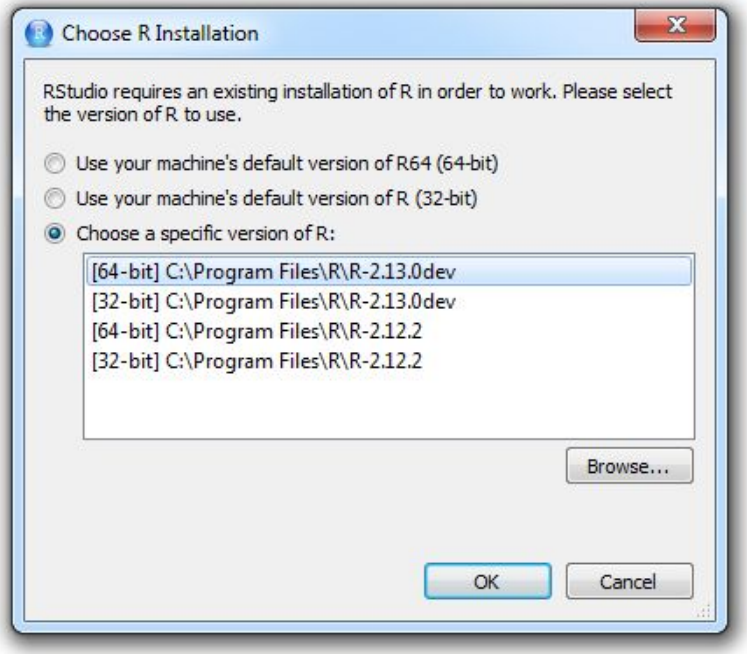
Instalação do Rstudio

Faculdade Multiversa

Download em https://www.rstudio.com/products/rstudio/download/

- Executar o arquivo baixado
- Selecionar o idioma
- Clicar em **próximo** nas telas seguintes

Obs: Quando R é instalado no Windows, ele grava a versão que está sendo instalada, no registro como a versão "atual" de R. Esta é a versão do R com a qual o RStudio é executado por padrão.



Minha primeira aplicação R...

Reza a lenda...

Escreva no editor do Rstudio a instrução abaixo e depois aperte Ctrl + Enter ou clique no botão **"Run"**

```
print("Alô mundo!!!")
```

O mesmo processo pode ser executado no console interativo, escreva a instrução e aperte Enter

Atribuição de valores

Execute as instruções abaixo

```
x = 2 # pode ser usada em variáveis globais e locais
y <- 4 # Deve ser usada somente em variáveis globais.
Z  = x/4*100
```

Como imprimir os valores de x, y e z no terminal?

Faculdade Multiversa

Operações Básicas no R

+	Adição	4+5
–	Subtração	7-2
/	Divisão	6*8
*	Multiplicação	9/3
^	Potência	4^3
sqrt()		**sqrt(25)**

Desafio: Crie variáveis para os guardar os resultados dos exemplos acima

Faculdade Multiversa

Introdução a linguagem R (comandos Básicos)

Principais operadores:

```
valor <- 0.83              # Atribuição de valores (Indicado para escopo global)
valor =  0.83              # Atribuição de valores geral
valor < 0.9                # Menor (> para maior)
valor <= 0.83              # Menor ou igual (>= para maior ou igual)
valor == 9                 # Igualdade
0 <= valor & valor <= 0.1 # Dois criterios aditivos
0 <= valor | valor <= 0.1   # Dois criterios, um ou outro
!valor == 0.83             # Inverter o argumento lógico
```

Introdução a linguagem R (comandos Básicos)

log(x = 8) # *Logaritmo natural de 8*

log(x = 8, base = 2) # Logaritmo de 8 na base 2, especificando cada argumento

rep(x = 1, times = 4) # *Repetir o valor 1 quatro vezes*

sum(1, 8, 79) # *Soma de vários valores*

As funções em R possuem basicamente a mesma estrutura:

nome da função(argumento1, argumento2…)

Faculdade Multiversa

Tipos primitivos

INTEIRO: Representa valores numéricos negativo ou positivo sem casa decimal, ou seja, valores inteiros.

REAL ou NUMÉRICO: Representa valores numéricos negativo ou positivo com casa decimal, ou seja, valores reais. Também são chamados de ponto flutuante.

LÓGICO: Representa valores booleanos, assumindo apenas dois estados, VERDADEIRO ou FALSO. Pode ser representado apenas um bit (que aceita apenas 1 ou 0).

TEXTO: Representa uma sequencia de um ou mais de caracteres, colocamos os valores do tipo TEXTO entre " " (aspas duplas) ou ' ' (aspas simples)

.

Verificando e modificando os tipos primitivos

```
class(Objeto) # mostra o tipo de objeto
var1 = 3

var2 =  3L

var3 =  3.0

class(var1)

class(var2)

class(var3)

class(as.integer(var1))
```

Por padrão o R salva qualquer valor numérico em seu tipo primitivo mais abrangente (numeric), para converter para o inteiro use a função as.interger().
Para converter inteiro para numérico ou real use a função as.numeric()

```
var4 = as.numeric(var2)

class(var4)
```

Verificando e modificando os tipos primitivos

```
class(Objeto) # mostra o tipo de objeto
var4 = "Olá mundo!"

var5 = TRUE

class(b)

class(c)
```

O que acontece se…?

```
var1 = 1
var2 = ‘2’
var3 = TRUE

print(var1 + 2)
print(var1 + var3)
print(var2 + 1) #erro
```

Como realizar a última operação acima sem reatribuir valores a var2?

Controle de fluxo de execução (Condicionais)

O R possui estruturas de if, else, for e while. Esses controles de fluxo são muito importantes na hora de programar, pois nos permitem manipular de modo eficiente as ações do computador.

```
idade = 18

if(idade<14){

 print('criança')

}
```

```
idade = 18if(idade<14){

 print('criança')

}else{

        print('jovem   ou
adulto')

 }
```

```
if(idade<14){

 print('criança')

}else if(idade>=14 & idade < 18
){

  print('jovem')

 }else{

  print('adulto')

 }
```

laço "for"

Controle de fluxo de execução

Laços ou repetições

Laço FOR

```
idade = 18

for ( i in c(1:10)) {

  print(paste0('Imprimido laço ',i))

}
```

Laço WHILE

```
a=1

while (a <=10 )  {

  print(paste0("Imprimindo laço ",a))

  a=a+1

}
```

Vetores

Os vetores são uma sequência simples de elementos do mesmo tipo. **Em uma tabela, cada coluna é um vetor.**

O vetor pode ser criado através da função c() ex:

```
vetor1 = c(2,4.2,5,4.88)  ## criando um vetor tipo numeric
vetor2 = c(FALSE, TRUE, TRUE) ## criando um vetor com valores lógicos (TRUE e FALSE)
vetor3 = c(NA, 9, 16, -1) ## NA indica que o valor é inexistente
vetor4 = rep(1,5) ## Vetor com valores repetidos utilizando a função rep() [rep(valor,repetições)]
vetor5 = rep("gato",3)   ## pode ser um vetor de "strings"
vetor6 = seq(1,2,0.2) ## Vetor com valores reais e espaçamento constante utilizando a função seq() [seq(início,fim,espaçamento)]
```

Data frames

Faculdade Multiversa

O Data Frame é a estrutura do R utilizada para armazenar elementos em forma de tabela, organizados em linhas e colunas. As colunas e linhas podem ser nomeadas. Pode ser criado através da função data.frame()

```
pacientes = data.frame(id = c("P1","P2","P3","P4"),
                       sexo = c("feminino", "feminino", "masculino", "masculino"),
                       peso=c(80, 85, 100, 95), idade = c(25, 32, 75, 61),
                       altura = c(150,191,170,156))
```

O R possui em seu pacote padrão alguns data frames que podem ser usados para testes e aprendizado:

```
print(mtcars)
```

```
rm(list = ls()) # apaga todos os objetos criados, use com moderação…
```

Faculdade Multiversa

Criando dataframes a partir de dados externos

As análises de dados geralmente utilizam dados externos, das mais diversas fontes e formatos de arquivos como:

- CSV - Comma-separated values;
- TXT - Arquivo texto;
- JSON - JavaScript Object Notation;
- XML - eXtensible Markup Language;
- TSV - Tab-separated values;
- XLS e XLSX - Arquivos do MS Excel;
- HTML - HyperText Markup Language;
- PDF - Portable Document Format
- Conexão com diversos SGBD (Oracle, PostgresSQL, Google BigQuery..)

Importando arquivo CSV

A função nativa **read.csv()** ou **read.csv2()** permite a importação de arquivos CSV, que é um formato que utiliza caracteres como vírgula, ponto e vírgula entre outros, para separação dos valores.
read.csv2(local_nome_do_arquivo, header = TRUE, sep = ";", dec = ",") além de indicar o nome e o local do arquivo, alguns parâmetros podem ser utilizados como:

- ➔ header = TRUE ou FALSE : indica se o arquivo tem cabeçalho
- ➔ sep = “” : indica qual caractere é usado como separador de valores
- ➔ dec = “” : indica qual caractere é usado como separador decimal
- ➔ codificação a ser assumida para strings de entrada. Ele é usado para marcar strings de caracteres como conhecidos em “Latin-1” ou “UTF-8”

Exemplo:

```
futebol = read.csv('dados/campeonato-brasileiro-estatisticas-full.csv')
```

Analisando o dataframe

Visualizando o dataframe:
View(futebol)
#Verificando a quantidade de registros ou linhas:
nrow(futebol)
# Gerando um resumo estatístico
summary(futebol)
# Qual o total de gols que ocorreram no campeonato
qtd_gols = sum(futebol$chutes_no_alvo)
print(qtd_gols)
# Qual a média de gols por partida?
qtd_partida = nrow(futebol)/2
media_gols = qtd_gols/qtd_partida
print(media_gols)

OBS: Uma coluna ou vetor pode ser acessado usando o caracter **$**

nome_dataframe$nome_variavel

A coletânea de pacotes Tidyverse

Faculdade Multiversa

O Tidyverse é uma coletânea de pacotes poderosos para a importação, manipulação e visualização de dados. Na verdade é mais do que isso, o Tidyverse é uma filosofia, uma ideia de como tratar os dados de uma forma eficiente e reproduzível. (Hadley Wickham et al. 2019a)

Quando carregamos o pacote *tidyverse* estamos carregando os seguintes pacotes: *dplyr, ggplot2, tidyr, readr, purrr, tibble, stringr, forcats.*

install.packages("tidyverse") # Instala o pacote *tidyverse*

library(tidyverse) #Carrega os pacotes

Fluxo tidyverse

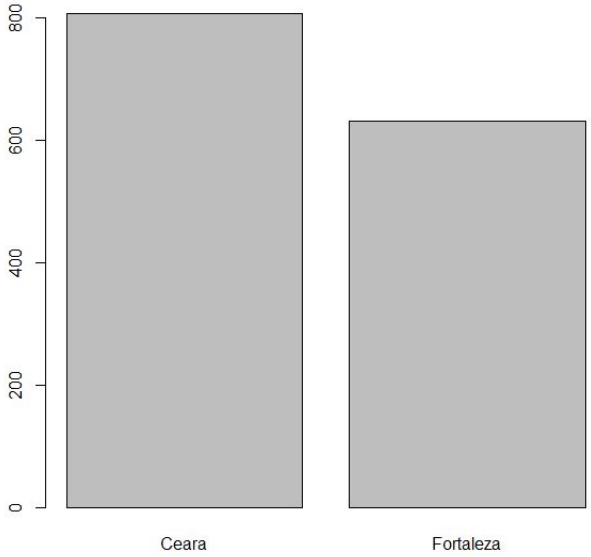
```
data_frame_2 <- data_frame %>%
 filter(
  #filtra o dataframe através de condicionais
      ) %>%
 mutate(
  #modifica e cria novas variáveis
 ) %>%
 select(
  #seleciona as variáveis necessárias para a análise
 ) %>%
 group_by(
  # cria uma estrutura de agrupamento entre variáveis
 ) %>%
 summarise(
   # realiza as estatísticas da análise e cria um novo quadro
de dados conforme as variáveis agrupadas )
```

O operador **%>%** (pipe) permite usar o valor resultante da expressão do lado esquerdo como primeiro argumento da função do lado direito, cria desta forma um fluxo de execução das funções.

Analisando com tidyverse

Faculdade Multiversa

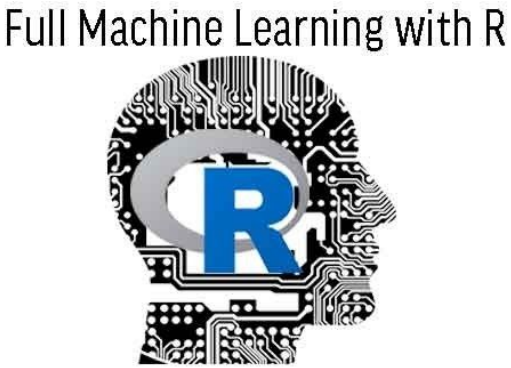
```
#Qual o quantitativo de partidas e gols  dos times cearenses
library(tidyverse)
resposta = futebol %>%
 filter(clube=='Ceara' | clube == 'Fortaleza') %>%
 group_by(clube) %>%
 summarise(qtd_partida = n_distinct(partida_id),
           qtd_gols = sum(chutes_no_alvo) )

#Qual a media de gols dos times cearenses
resposta = resposta %>%
 mutate(media_gols = qtd_gols/qtd_partida)

# Gerando um gráfico simples com a resposta:
clubes = unique(resposta$clube)
barplot(resposta$qtd_gols,names.arg = resposta$clube)
```



O que o R tem mais a oferecer?

Criação de documentos online: https://livro.curso-r.com/

Criação de dashboards interativos:
https://testing-apps.shinyapps.io/flexdashboard-shiny-crandash/



Fonte de pesquisa:

https://www.cetax.com.br/blog/

https://ceweb.br/guias/dados-abertos/capitulo-35/

https://felipegalvao.com.br/pt/blog/basic-r-introduction-data-types-and-structures/

https://www.lampada.uerj.br/arquivosdb/_book2/

https://r4ds.had.co.nz/index.html

https://escoladedados.org/tutoriais/mas-o-que-significa-isso-introducao-a-analise-de-dados/

Material do curso:

https://github.com/professorestudante/curso_r_2023

Faculdade Multiversa

Obrigado!!!