

PROGRAMAÇÃO EM R

AULA 01 – A LINGUAGEM R: UMA INTRODUÇÃO E SEUS FUNDAMENTOS

AGENDA

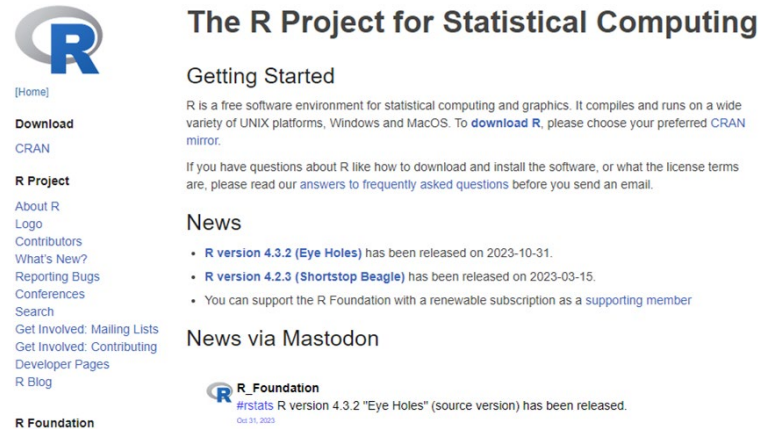
- Apresentação do curso;
- Introdução;
- Fundamentos do R.

O CURSO PROGRAMAÇÃO EM R

- Apresentação do curso;
 - R
 - A Linguagem R: uma introdução e seus fundamentos;
 - Manipulação de dados no R: da carga ao processo de transformação;
 - A construção de gráficos com a linguagem R;
 - Criando um projeto completo no R.
- Objetivo: construir uma base inicial para aplicação da linguagem R em outras disciplinas;
- Bibliografia;
 - Wickham, H., & Grolemund, G. (2016). R for data science: import, tidy, transform, visualize, and model data. "O'Reilly Media, Inc."
- Calendário e horário;
 - 4 aulas
 - 19:30-22:50 (20 min de intervalo)
- Avaliação
 - Média das listas de exercícios;
 - Grupos de 3-4 alunos, distribuídos de forma aleatória.

INTRODUÇÃO




















■ O que é a Linguagem R?



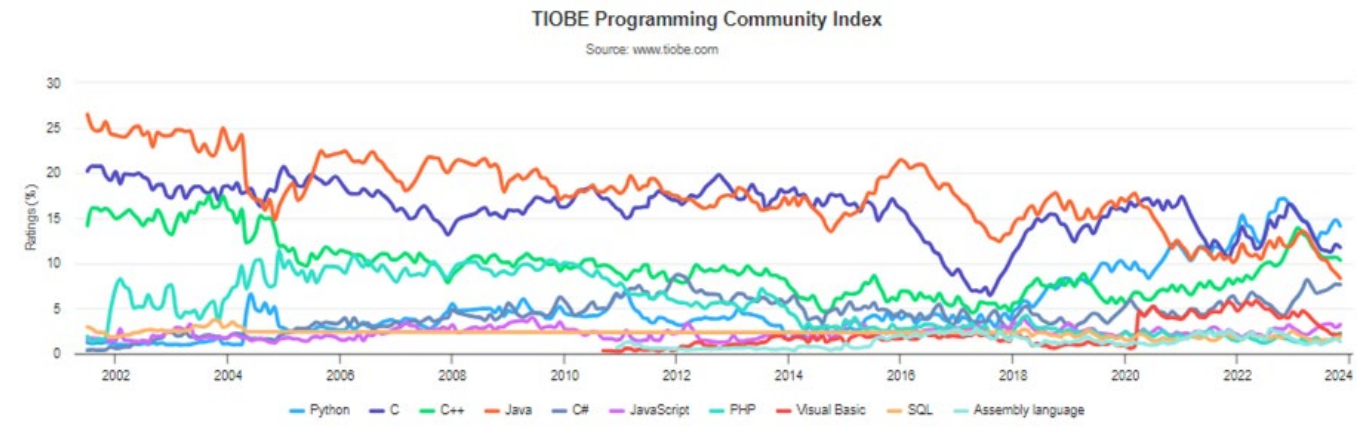
- A linguagem R nasceu durante a década de 90, inicialmente como um projeto de pesquisa de Ross Ihaka e Robert Gentleman.
- A linguagem R não é uma linguagem de programação completa. Ela apresenta algumas características de linguagem de programação:
 - Variáveis;
 - Estruturas de controle: condicionais e loops;
 - Funções.
- É a linguagem mais recomendada para análise estatística.

INTRODUÇÃO

Rankings Linguagem de Programação

Nov 2023	Nov 2022	Change	Programming Language	Ratings	Change
1	1		 Python	14.16%	-3.02%
2	2		 C	11.77%	-3.31%
3	4	▲	 C++	10.36%	-0.39%
4	3	▼	 Java	8.35%	-3.63%
5	5		 C#	7.65%	+3.40%
6	7	▲	 JavaScript	3.21%	+0.47%
7	10	▲	 PHP	2.30%	+0.61%
8	6	▼	 Visual Basic	2.10%	-2.01%
9	9		 SQL	1.88%	+0.07%
10	8	▼	 Assembly language	1.35%	-0.83%
11	17	▲	 Scratch	1.31%	+0.43%
12	24	▲	 Fortran	1.30%	+0.74%
13	11	▼	 Go	1.19%	+0.05%
14	15	▲	 MATLAB	1.15%	+0.14%
15	28	▲	 Kotlin	1.15%	+0.68%
16	14	▼	 Delphi/Object Pascal	1.14%	+0.07%
17	18	▲	 Swift	1.04%	+0.17%
18	19	▲	 Ruby	0.99%	+0.14%
19	12	▼	 R	0.93%	-0.20%
20	20		 Rust	0.91%	+0.16%

TIOBE Programming Community Index



INTRODUÇÃO

■ Vantagens e Desvantagens do R



Gratuita
Open source
Grande comunidade
Grande variedade de
pacotes disponíveis
**Excelente para análise
de dados**
Flexível e personalizável
Rapidez



Não há interface gráfica
**Limitações de
desempenho com
grandes datasets**
Curva de aprendizado
íngreme

INTRODUÇÃO

- A matéria prima para análise: o dado e sua progressão



INTRODUÇÃO

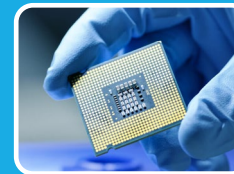
- Tempestade perfeita: um cenário favorável para a área de ciência de dados



Crescimento exponencial do volume de dados



Preço baixo de armazenamento de dados



Aumento significativo da capacidade de processamento dos computadores

INTRODUÇÃO

■ Data science e a evolução dos sistemas analíticos

- O que ocorreu?

**Análise
Descritiva**



- Por que ocorreu?

**Análise
Diagnóstica**



- O que ocorrerá?

**Análise
Preditiva**



- O que fazer?

**Análise
Prescritiva**



Passado

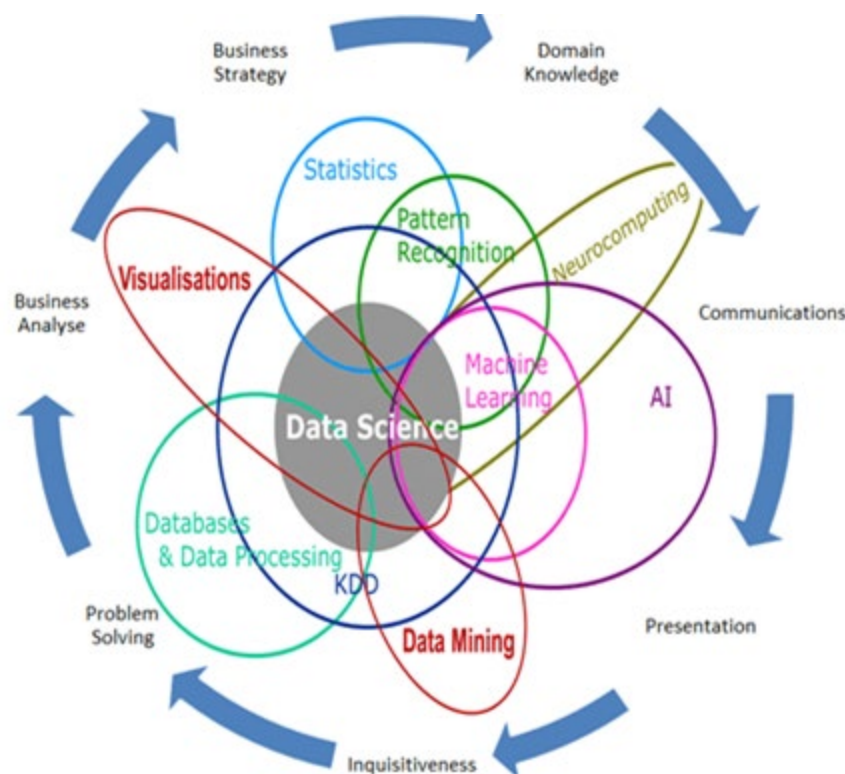
Futuro

BI Tradicional

Data Science

INTRODUÇÃO

- Data Science: uma área multidisciplinar.



A ciência de dados é uma disciplina que se concentra na extração de insights significativos de dados. É uma área interdisciplinar que combina princípios e práticas de matemática, estatística, inteligência artificial e engenharia da computação. É "o processo de usar dados para entender o mundo ao nosso redor e tomar decisões informadas".

Tierney B. (2012)

INTRODUÇÃO

- Ciclo de vida de um projeto de ciência de dados: modelo KDD

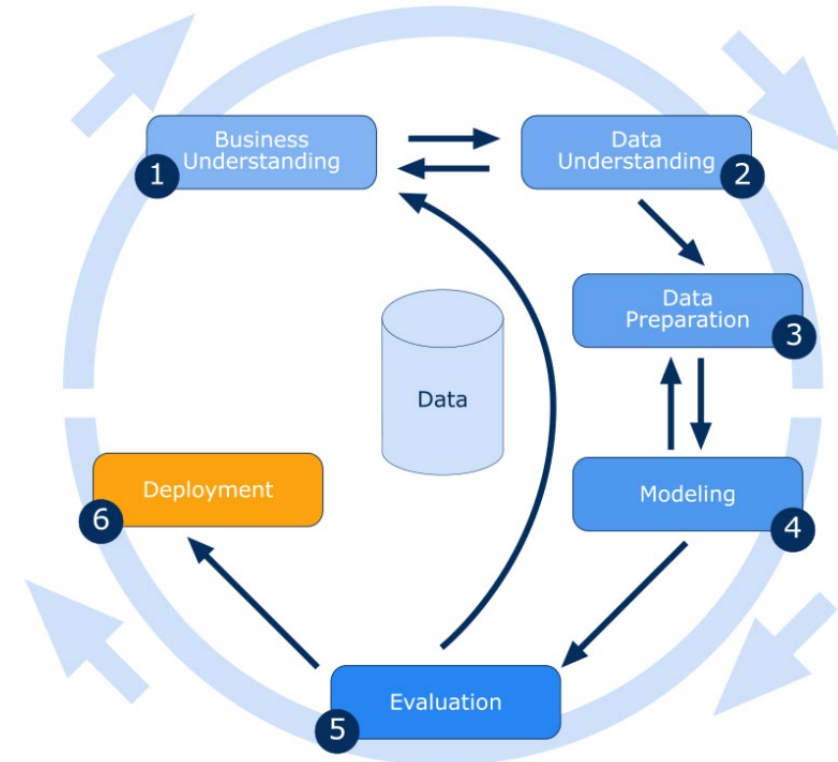


Fonte: tradução adaptada de (FAYYAD et al. 1996a)



INTRODUÇÃO

- Ciclo de vida de um projeto de ciência de dados: modelo CRISP-DM



FUNDAMENTOS DA LINGUAGEM R

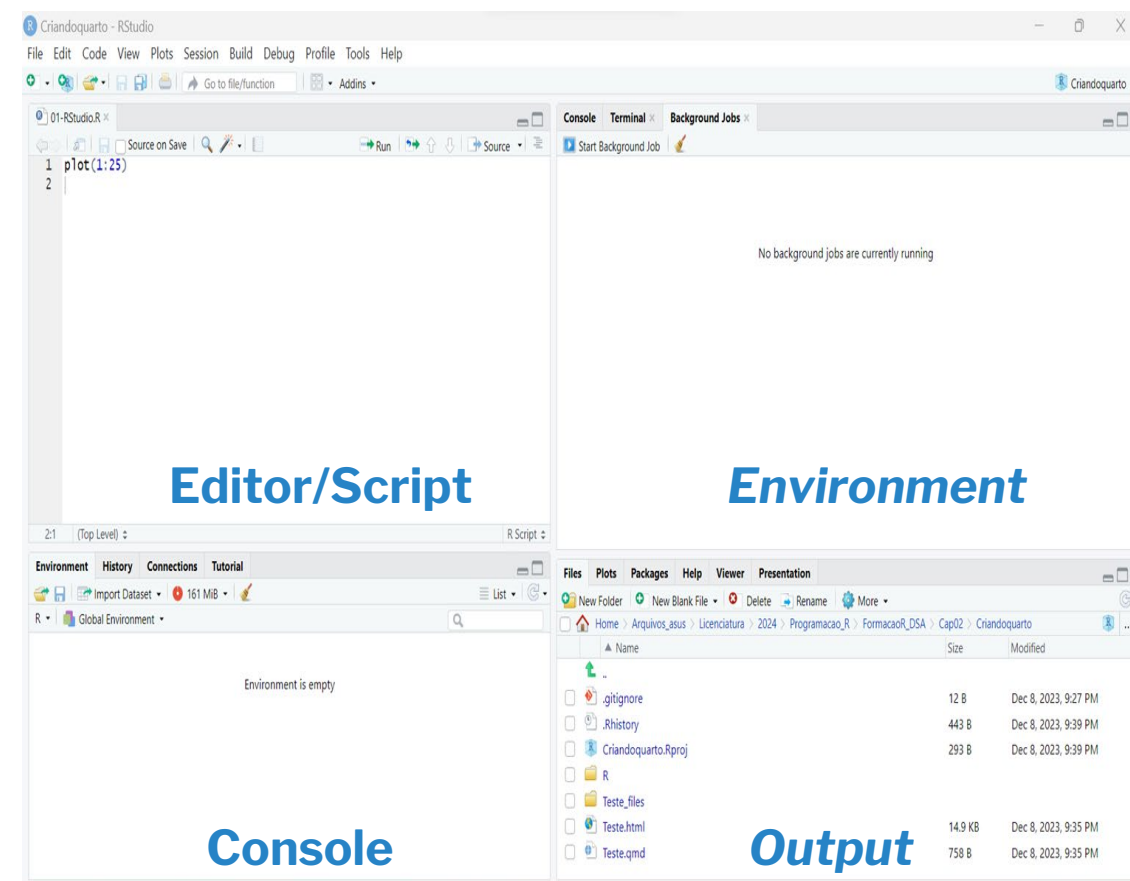
NOÇÕES BÁSICAS

- R é uma linguagem e um programa para interpretar os códigos em linguagem R;
- O RStudio é um ambiente de desenvolvimento integrado (IDE) para programação em R, funcionando como um facilitador;
- O RStudio não funciona sem o R, pois é o último que interpreta os códigos e devolve os resultados;
- É possível utilizar o RStudio instalado na máquina e na nuvem (POSIT);
- RStudio possui 4 blocos principais: Editor/Scripts, Console, Environment e Output.

FUNDAMENTOS DA LINGUAGEM R

NOÇÕES BÁSICAS

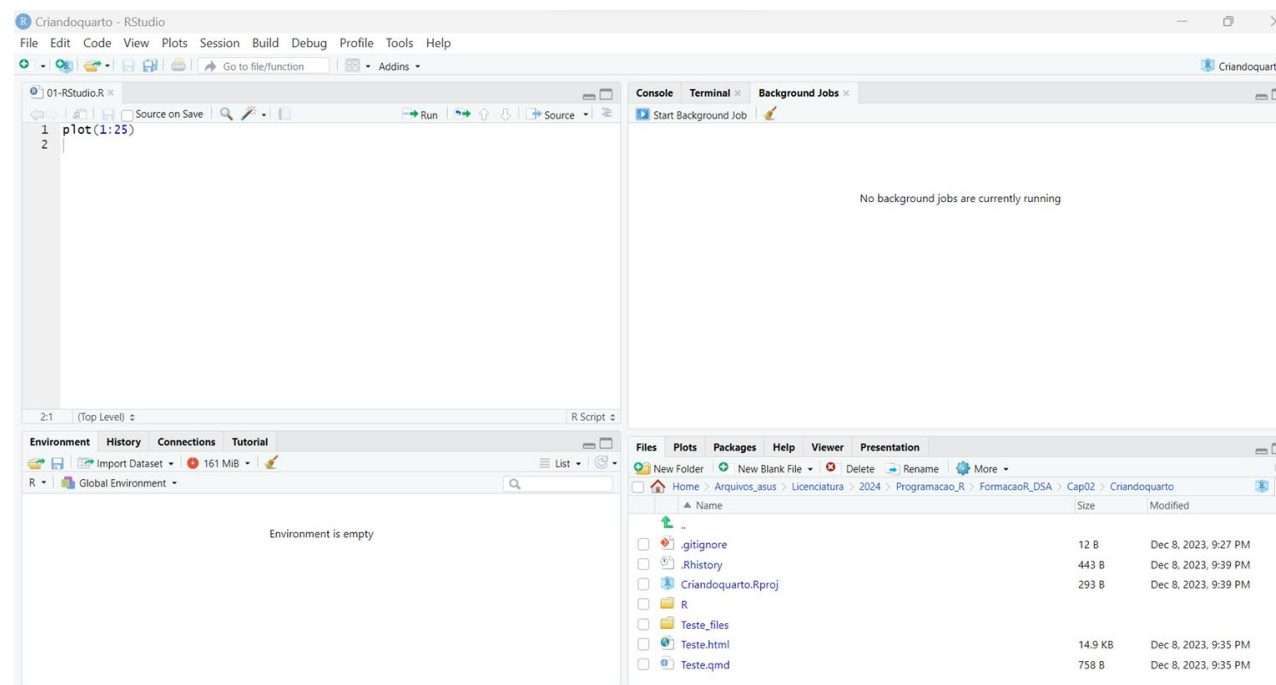
- Detalhando os blocos principais:
 - **Editor** : painel onde escrevemos o código;
 - **Console**: painel onde rodamos o código;
 - **Environment**: painel que apresenta todos os objetos criados;
 - History: painel com o histórico de comandos;
 - **Output**
 - Files: mostra os arquivos do computador;
 - Packages: evidencia os pacotes instalados e carregados;
 - Plots: painel onde os gráficos são mostrados;
 - Help: janela de documentação e ajuda
 - Viewer: painel onde relatórios e dashboards serão apresentados.



FUNDAMENTOS DA LINGUAGEM R

NOÇÕES BÁSICAS

- Funcionalidades importantes:
 - **Vassourinha dos quadrantes:** limpeza;
 - **Comentário:** símbolo # antes;
 - **Set Working Directory:**
 - `setwd()`: mudar o diretório de trabalho;
 - `getwd()`: mostrar o diretório de trabalho;
 - `dir()`: listar o conteúdo do diretório de trabalho;
 - **Pacotes:**
 - `install.packages()`: instalar pacote;
 - `library()`: carregar pacote;
 - **Observações relevantes:**
 - Case sensitive;
 - Separador decimal: padrão é o ponto;
 - Caracteres especiais: devem ser evitados.



FUNDAMENTOS DA LINGUAGEM R

NOÇÕES BÁSICAS

- Funcionalidades importantes:

- **Ajuda:**

- **help():** colocar nome da função;

- ?

- **Pacote sos:** função findFn();

- **help.search():** pesquisa quando não sabe o nome da função;

- ??

- **Rsitesearch():** busca no site do R em toda sua documentação;

- **example():** apresenta um exemplo de uso da função;

- **Atribuição de objetos**

- <- ou =;

- **Mostrar objetos**

- **ls() ou objects():** listar objetos;

- **rm() :** remover objetos;

- **Símbolos especiais**

- NA, NAN, Inf, TRUE, FALSE, NULL, pi;

TIPOS DE OPERADORES

■ Operadores Básicos

- Soma: +
- Subtração: -
- Multiplicação: *
- Divisão: /
- Potência: ^
- Módulo (resto divisão): %%

■ Operadores Relacionais

- Atribuição de variáveis: = ou <-
- Operadores: >, <, >=, <=, ==, != (diferente)

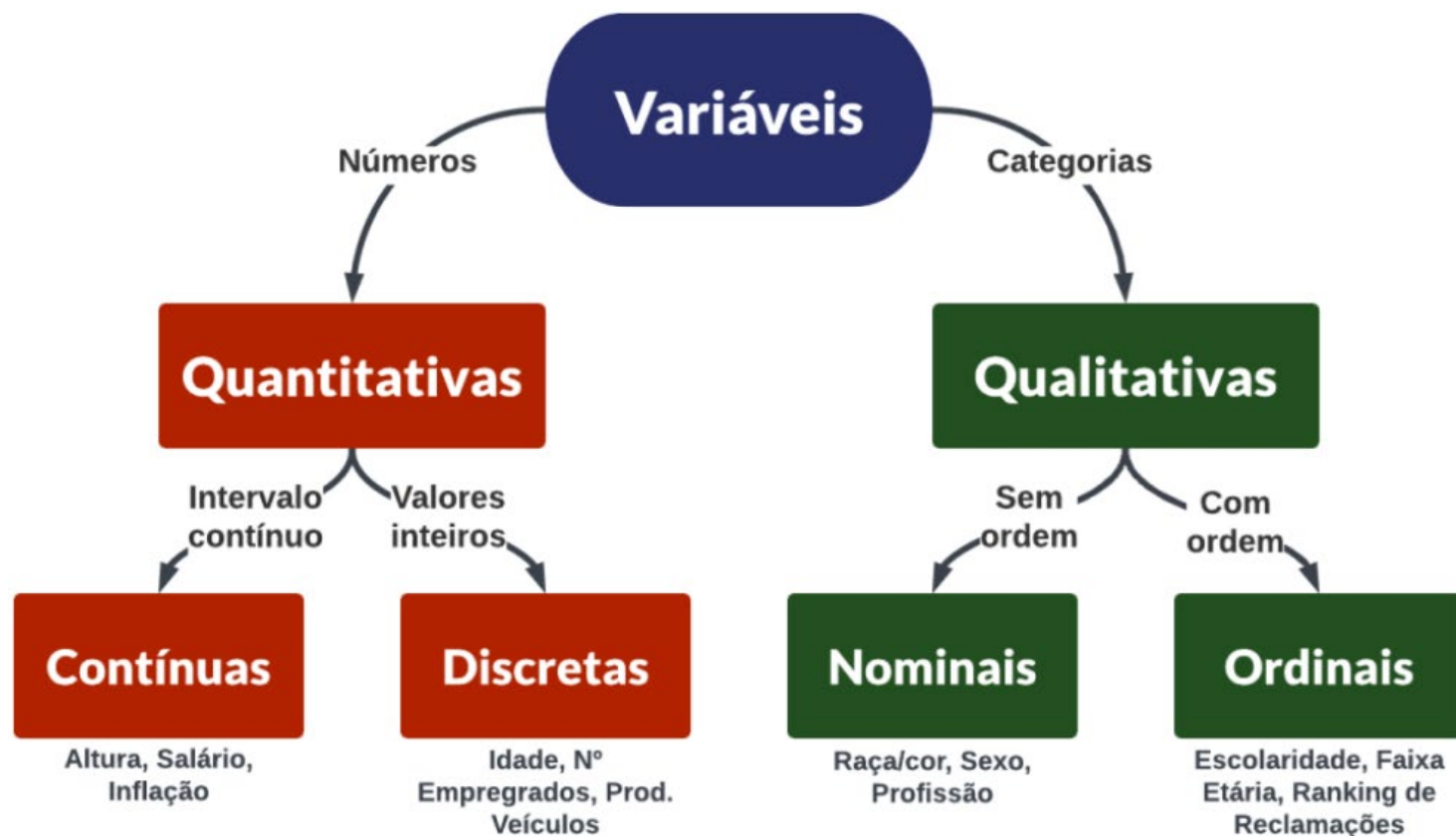
■ Operadores Lógicos

- E: &
- Ou: | (símbolo pipe)
- Negação: !

VARIÁVEIS/ OBJETOS

- **Variável é uma área em memória onde o computador armazena dados;**
- **Criar uma variável;**
 - `var1 = 10;`
 - `var2 = 2;`
- **Definir uma variável a partir de outra variável;**
 - `var1 = var2;`
- **Variável como uma lista de elementos;**
 - `var3 = c("a","e","i","o","u");`
- **Variável como uma função;**
 - `var4= function(x) {x+1};`
- **Informações sobre as variáveis:**
 - `class();`
 - `typeof().`

TIPOS DE VARIÁVEIS



TIPO DE DADOS

■ Numérico (numeric)

- Por padrão o R cria uma variável como numérica e com o tipo double (decimal);
- `as.integer()`: converter uma variável para inteiro;

■ Caracter (character)

- Pode ser um carácter ou um conjunto de caracteres;

■ Lógico (logical)

- TRUE e FALSE.

FUNDAMENTOS DA LINGUAGEM R

TIPO DE ESTRUTURA DE DADOS

No R podemos armazenar nossos dados das seguintes formas:

- Vetor;
- Fator;
- Matriz;
- Array;
- Lista;
- Dataframe;
- TS (time series).

TIPO DE ESTRUTURA DE DADOS

■ Vetor

- 1 dimensão e 1 tipo de dado;
- Armazenamento de um ou mais elementos;
- **Funções:**
 - **c():** criar;
 - **length():** comprimento;
 - **names():** nomear cada elemento;

■ Fator

- Fatores são uma classe de objetos no R criada para representar e armazenar as variáveis categóricas numericamente, garantindo maior performance de processamento;
- Cada categoria única é armazenada somente 1x e os dados são armazenados como um vetor de inteiros;
- As categorias podem ou não serem ordenadas;
- 1 dimensão e 1 tipo de dado;
- **Funções:**
 - **factor:** criar um vetor como fator;
 - **levels():** apresenta as categorias;
 - **nlevels():** número de categorias.

FUNDAMENTOS DA LINGUAGEM R

TIPO DE ESTRUTURA DE DADOS

■ Matriz

- 2 dimensões (linhas e colunas) e 1 tipo de dado;
- **Funções:**
 - **matrix():** criar;
 - **dim(), nrow(), ncol():** tamanho da matriz, número de linhas e colunas respectivamente;
 - **rownames() e colnames:** nomear linhas e colunas;

■ Array

- 2 ou mais dimensões e 1 tipo de dado;
- **Funções:**
 - **array():** criar;

TIPO DE ESTRUTURA DE DADOS

■ Tabela de dados – Data frames

- Matriz com diferentes tipos de dados;
- **Funções:**
 - **data.frame():** criar;
 - **str():** resumo sobre a tabela;
 - **dim(), nrow(), ncol():** tamanho do dataframe, número de linhas e colunas respectivamente;
 - **head():** apresenta n primeiras linhas;
 - **tail():** apresenta n últimas linhas;
 - **rownames() e colnames:** nomear linhas e colunas;

■ Lista - list

- Coleção de diferentes tipos de objetos podendo ter diferente tipos de dados;
- **Funções:**
 - **str():** resumo da estrutura da lista;
 - **length():** comprimento da lista;
 - **names():** atribuir nomes a lista.

FUNDAMENTOS DA LINGUAGEM R

OPERAÇÕES COM VETORES

- **Indexação[]:** índice dos elementos dentro do vetor;
- **Combinação de vetores:** `c()`;
- **Operações matemáticas com vetores;**
- **Operações com vetores com diferentes elementos;**
- **Nomear vetores:** `names()`.

OPERAÇÕES COM MATRIZES

- **Criar matrizes a partir de número de linhas ou colunas;**
- **Indexação[,]:** com 2 elementos;
- **Matriz transposta:** `t()`;
- **Matriz inversa:** `solve()`;
- **Nomear matrizes:** `dinames()`;
- **Combinar matrizes:** `rbind()` e `cbind()`.

OPERAÇÕES COM LISTAS

- **Indexação** `[] []`;
- **Nomear listas**: `names()`;
- **Ao nomear a lista pode chamar os elemento com \$**: `lista$caracteres`;
- **Unir objetos diferentes**: `list()`.

FUNDAMENTOS DA LINGUAGEM R

OPERAÇÕES COM DATAFRAMES

- **Criando dataframe a partir de vários vetores;**
- **Indexação[,]:** várias maneiras;
- **Filtro a partir de uma regra;**
 - O filtro não altera o dataframe, é somente uma seleção;
- **Summary():** resumo estatístico de cada variável
- **Combinar dataframes:** merge().

- Temos aqui alguns atalhos importantes utilizados no R:
 - **CTRL+ENTER:** roda a(s) linha(s) selecionada(s) no script. O atalho mais utilizado;
 - **ALT+-:** cria no script um sinal de atribuição (<-). Você o usará o tempo todo;
 - **CTRL+SHIFT+M:** (%>%) operador pipe;
 - CTRL+1: altera cursor para o script;
 - CTRL+2: altera cursor para o console;
 - ALT+SHIFT+K: janela com todos os atalhos disponíveis.

REFERÊNCIAS E LINKS

- Cientistas de Dados no Github:
 - <https://github.com/prakhar1989>
 - <https://github.com/wesm>
 - <https://github.com/jakevdp>
 - <https://github.com/mblondel>
 - <https://github.com/mnielsen>
 - <https://github.com/jtleek>
 - <https://github.com/allisonhorst>
 - <https://github.com/jbrownlee>
- Sites
 - <http://www.datasciencecentral.com>
 - <http://www.kdnuggets.com>
 - <http://www.predictiveanalyticstoday.com>
 - <http://www.cienciaedados.com>
 - <http://www.r-bloggers.com>
 - <https://rpubs.com>
 - <https://machinelearningmastery.com/blog/>
 - <https://stackoverflow.com/>
 - <https://medium.com/>
 - <https://towardsdatascience.com/>
 - <https://www.datacamp.com/>