MBA USP ESALQ

Estruturas de Bancos de Dados, Tipos de Variáveis e Escalas de Mensuração

Introdução ao Machine Learning

Rafael de Freitas Souza

Data Science

Conjunto de técnicas de programação voltadas à coleta, ao tratamento, à manipulação, à organização, à análise, à extração de informação e à apresentação de dados, na forma de relatórios ou gráficos, visando subsidiar o processo de tomada de decisão.

Técnicas de Data Science a serem abordadas durante o Módulo em curso:

Introdução ao *data wrangling*;

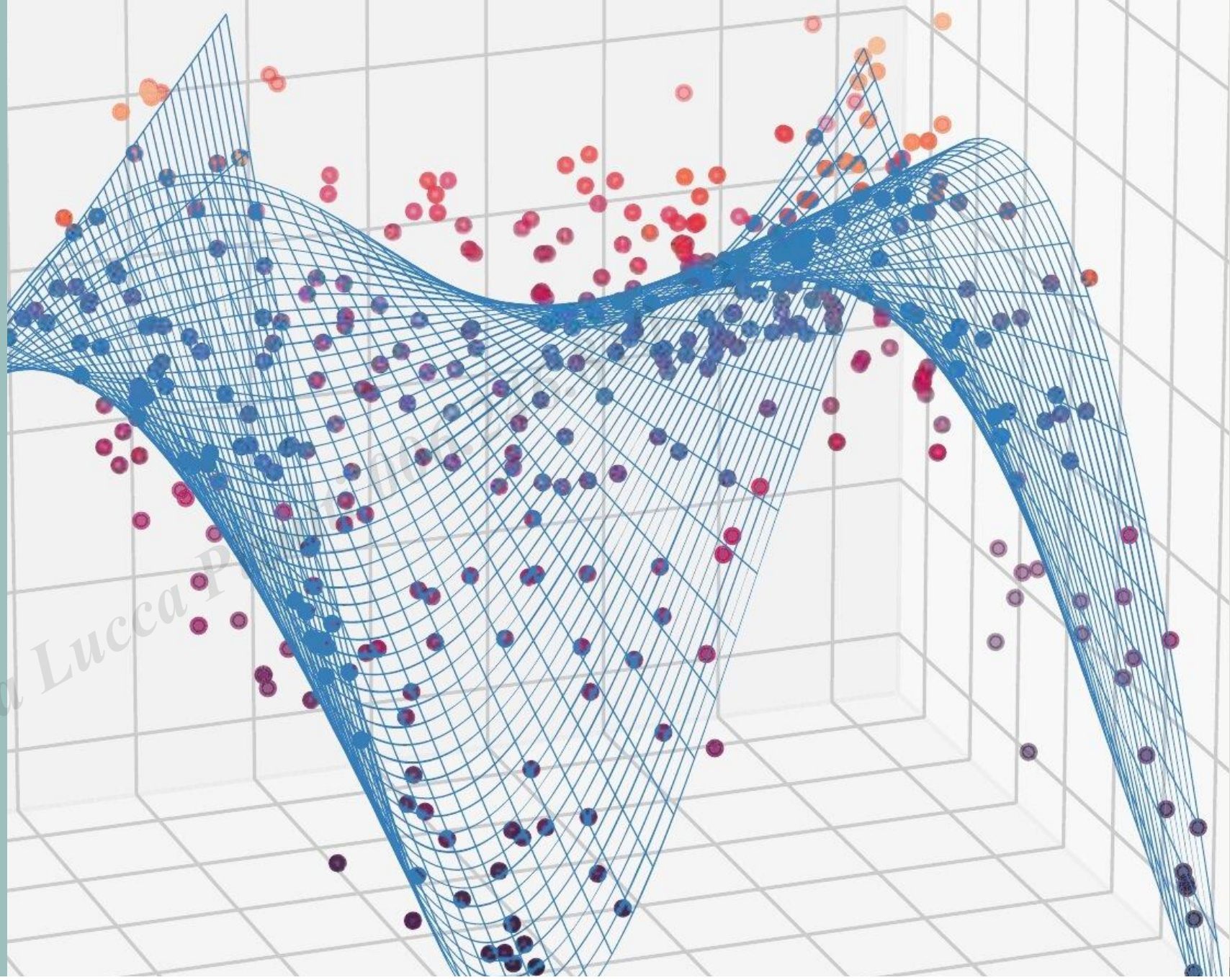
Construção e estruturação de *datasets*;

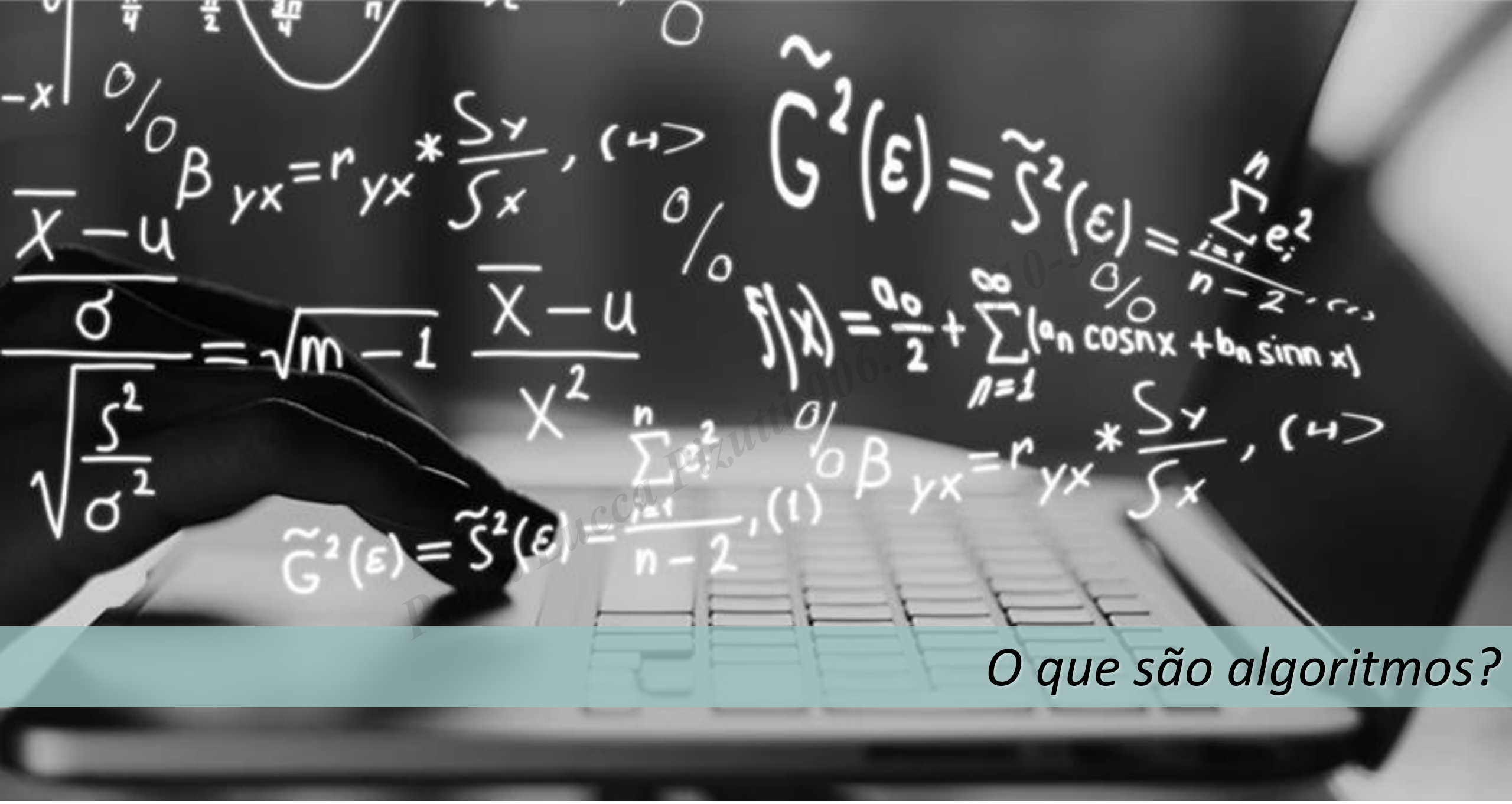
Escalas de mensuração de variáveis;

Introdução à linguagem computacional R;

Algoritmos de *machine learning*, compreendendo:

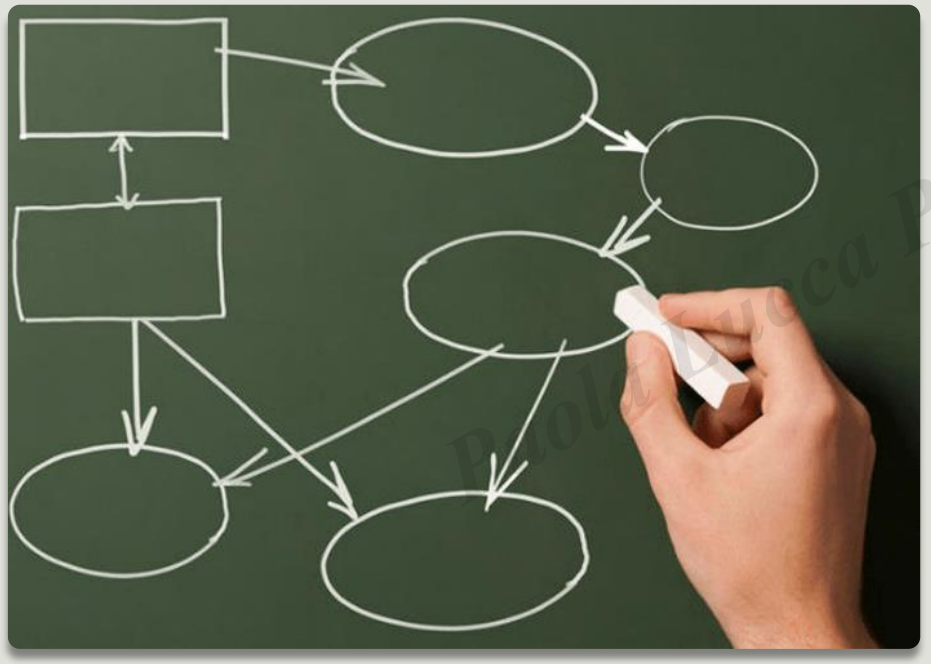
- Algoritmos não supervisionados – técnicas exploratórias;
- Algoritmos supervisionados – técnicas preditivas.





O que são algoritmos?

Algoritmos – um conceito:



Algoritmos são sequências explícitas, literais, limitadas e sistêmicas de instruções e de operações direcionadas à consecução de um dado objetivo pré-definido.

Basicamente, qualquer verbo conhecido, desde que denote uma ação destinada a humanos, pode ser considerado um algoritmo.

Link: https://www.youtube.com/watch?v=pdhqwbUWf4U

Taxonomia Básica de Espécies de Algoritmos de *Machine Learning*

NÃO SUPERVISIONADOS

Técnicas de *machine learning* assentadas com algoritmos **não supervisionados** (*unsupervised learning*) não possuem capacidade de inferência. Dedicam-se, portanto, a uma análise exploratória, diagnóstica, de dado fenômeno estudado. Exemplos comuns: Análise de *Clusters*, Análise Fatorial de Componentes Principais, Análises de Correspondências Simples e Múltiplas, etc.

SUPERVISIONADOS

Técnicas de *machine learning* assentadas com algoritmos **supervisionados** (*supervised learning*) possuem capacidade de inferência. Dedicam-se, portanto, a uma análise confirmatória, preditiva, de dado fenômeno estudado. Exemplos comuns: Regressões Lineares, Regressões Logísticas, Árvores de Decisão, *Random Forests*, Redes Neurais, etc.


Como escolher os algoritmos?

Passo 1: Algoritmos Não Supervisionados ou Algoritmos Supervisionados?

O primeiro passo é a definição do problema que se quer resolver, seja ele acadêmico ou não.

- Se há objetivos de **fazer inferências** para observações não presentes na amostra que foi utilizada para o treino do algoritmo, **o ideal é a utilização de algoritmos supervisionados**;
- Se há objetivos de **fazer diagnósticos**, sem a intenção de fazer inferências para observações não presentes na amostra que foi utilizada para o treino do algoritmo, **o ideal é a utilização de algoritmos não supervisionados**.

Passo 2: A Construção e a Estruturação de uma Base de Dados

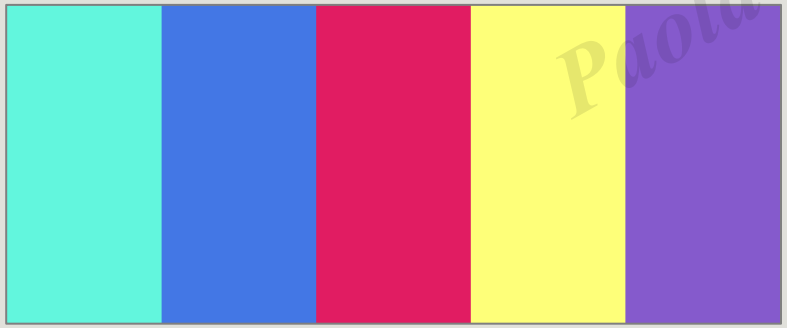
Regra geral, as bases de dados são estruturadas da seguinte maneira: variáveis em colunas e observações em linhas.

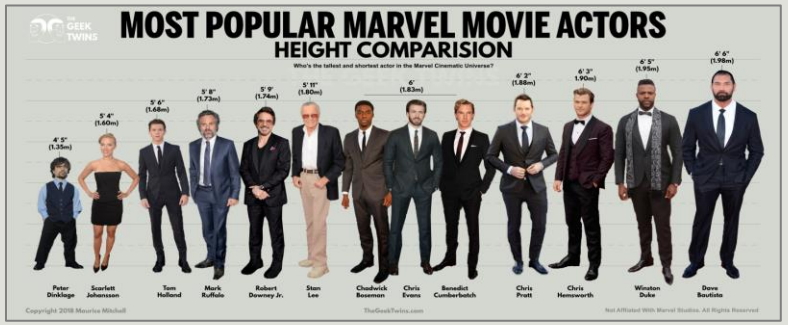
id	palavras suspeitas	remetente desconhecido	presença de imagens	classificação
1	sim	não	sim	spam
2	sim	sim	não	spam
3	sim	sim	não	spam
4	não	sim	sim	genuíno
5	não	não	não	genuíno
6	não	não	não	genuíno

Passo 3: Quais as Escalas de Mensuração de suas Variáveis?

A definição incorreta das escalas de mensuração das variáveis da base de dados é um dos principais erros na aplicação das técnicas de *machine learning.* Tal erro é irreparável, implicando no reinício de todo o processo de modelagem, em razão dos vieses criados (e.g.: a ponderação arbitrária).

Em suma: suas variáveis são apenas quantitativas, apenas qualitativas ou há a presença dos dois tipos?





O que é uma variável?

O que é uma variável?

Variáveis podem ser entendidas como uma característica de dada amostra ou população, que pode ser medida, ou contada ou categorizada.

São bons exemplos introdutórios, a altura e/ou o peso das pessoas, suas faixas de renda, a cor e/ou o modelo de carros que dirigem.

Os indivíduos de uma dada amostra ou população, não necessariamente, precisam ser pessoas em seu sentido físico. Podem ser objetos, distritos, municípios, organizações, grupos, células, moléculas, astros, etc. Dessa forma, as características dos indivíduos mencionados, seriam consideradas suas variáveis.

Para o curso, estabeleceremos a escala de mensuração das variáveis em duas: i) variáveis qualitativas; e ii) variáveis quantitativas.

Variáveis Qualitativas

- Também são conhecidas como variáveis latentes ou variáveis categóricas. São variáveis que não podem ser medidas; tão somente, categorizadas ou contadas.
- Por não poderem ser medidas, não permitem o cálculo de estatísticas descritivas de posição – e.g.: a media e a mediana.
- Por outro lado, podemos estabelecer tabelas de frequências para as suas categorias.
- Dividem-se em categóricas nominais e categóricas ordinais.

Variáveis Quantitativas

- Também conhecidas como variáveis métricas, ao contrário das variáveis qualitativas, as variáveis quantitativas podem ser mensuradas, possuindo, por óbvio, uma respectiva unidade de medida.

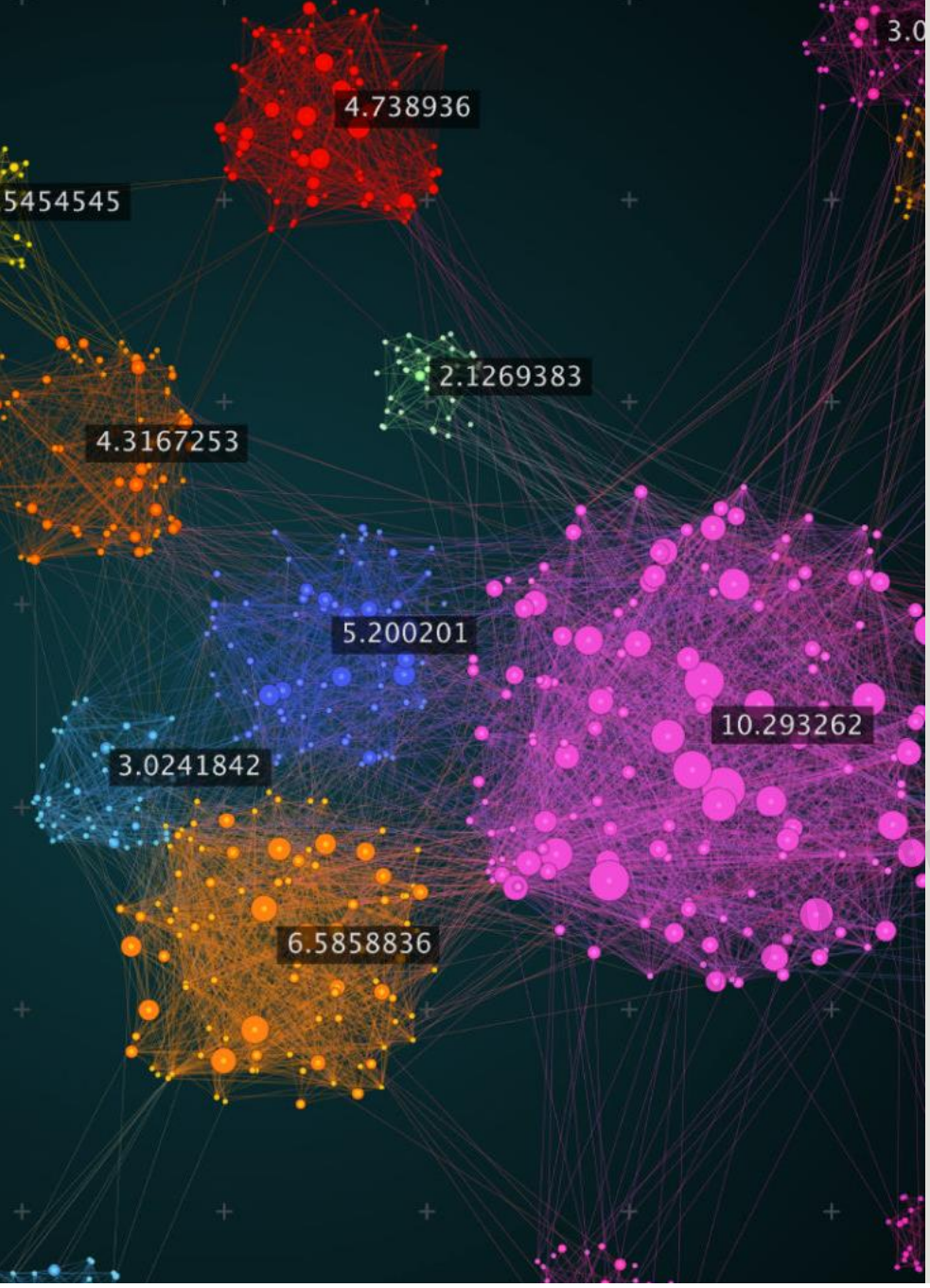
- Permitem o cálculo da média e da mediana, por exemplo.

- Dividem-se em variáveis contínuas e variáveis discretas.

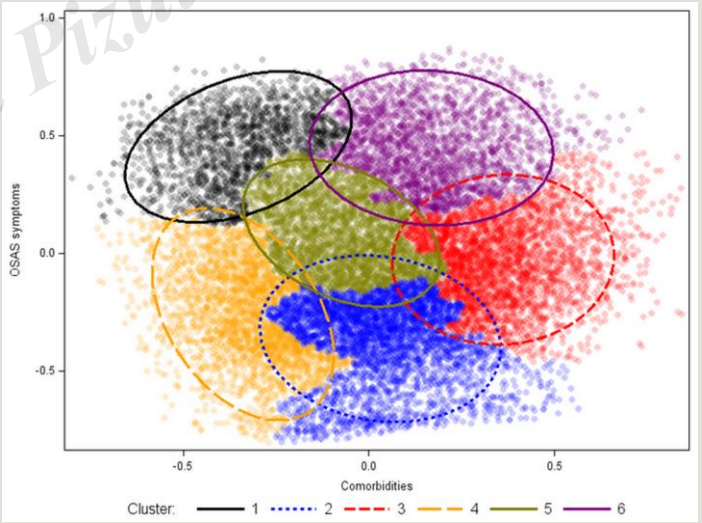
Tipos de algoritmos de machine learning que serão estudados

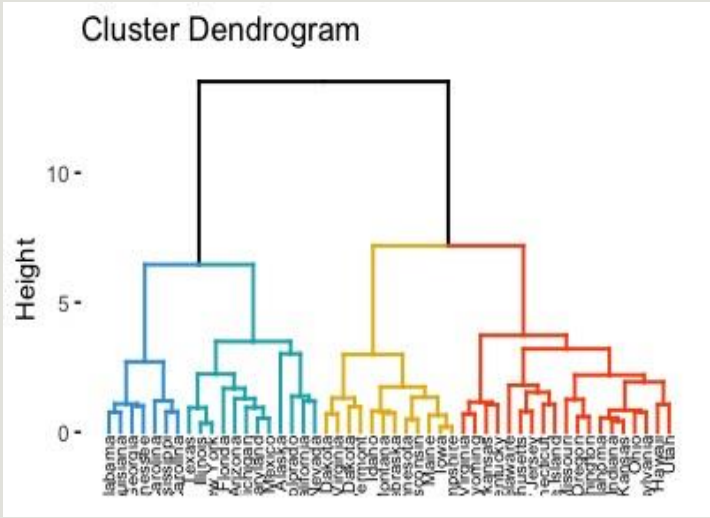
Algoritmos Não Supervisionados

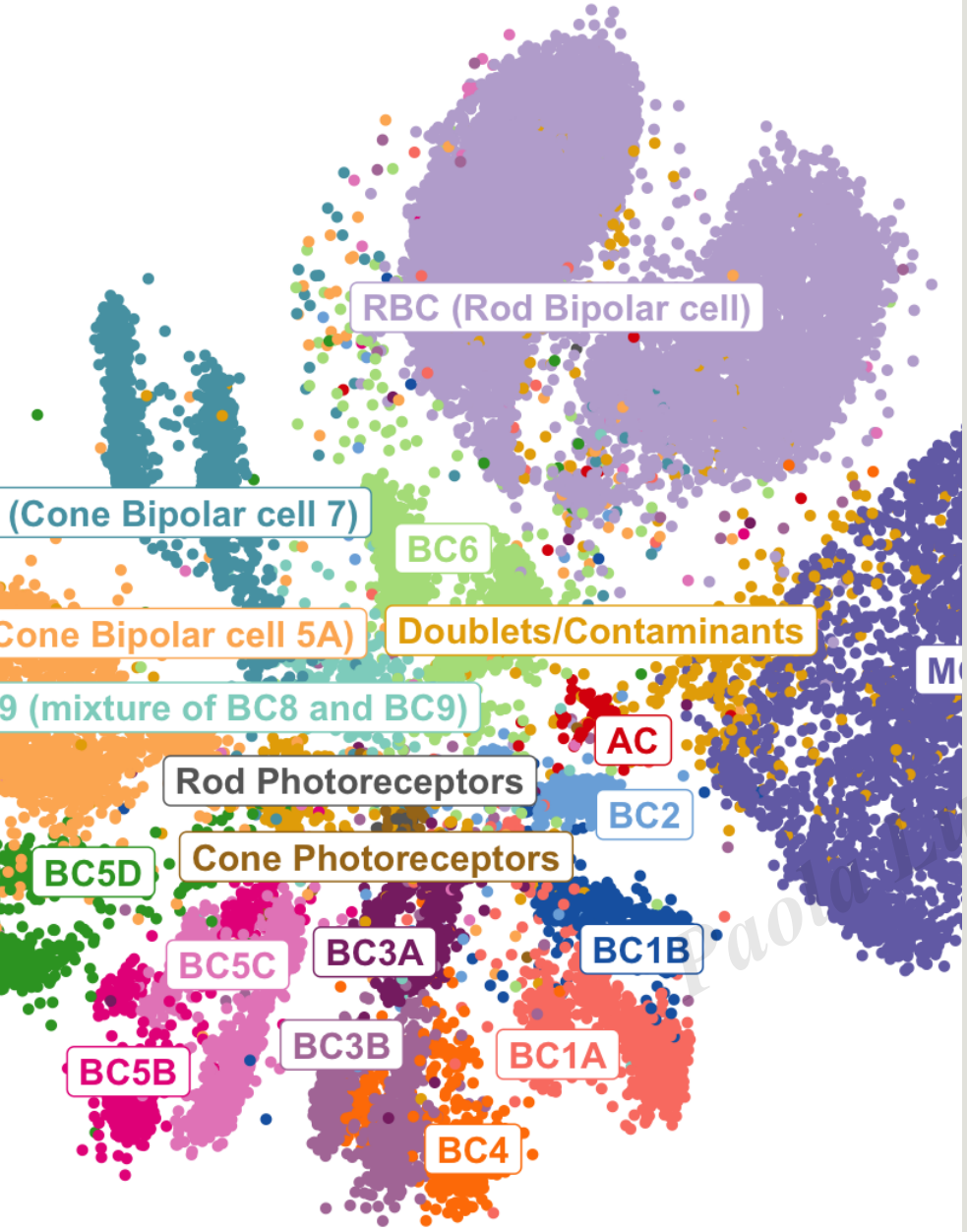
Análise de Agrupamentos	▪ Variáveis métricas (medidas de distância); ▪ Variáveis binárias (medidas de semelhança).
Análise Fatorial por Componentes Principais	▪ Variáveis métricas.
Análises de Correspondências Simples e Múltiplas	▪ Variáveis categóricas.



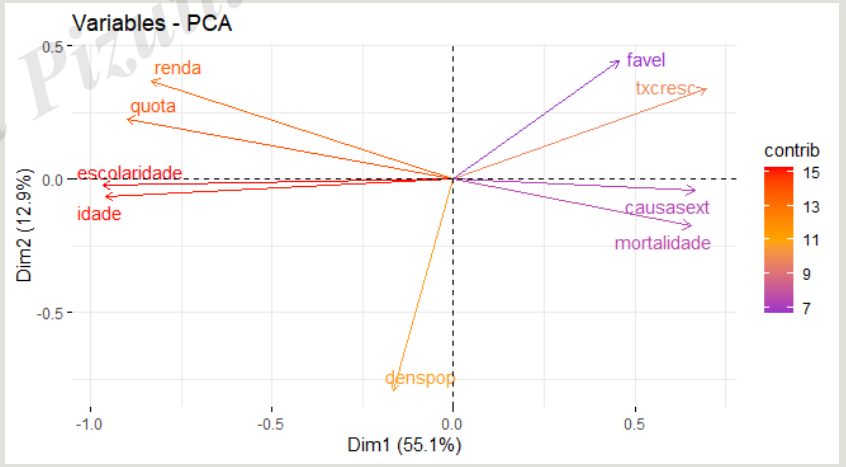
Análise de Agrupamentos

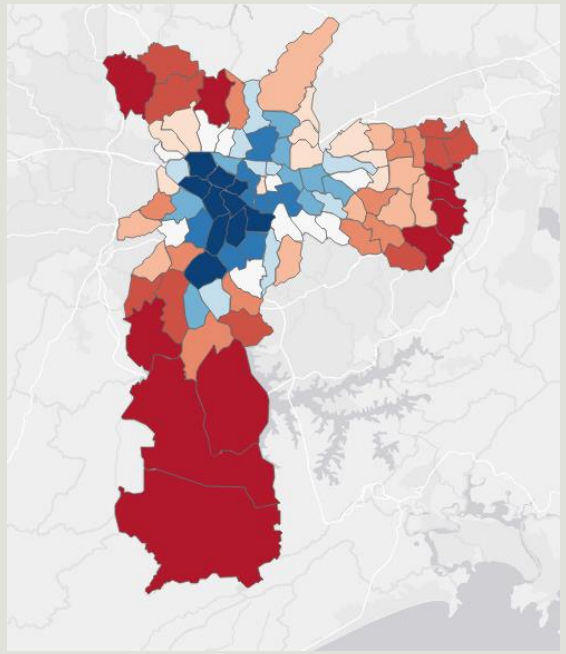


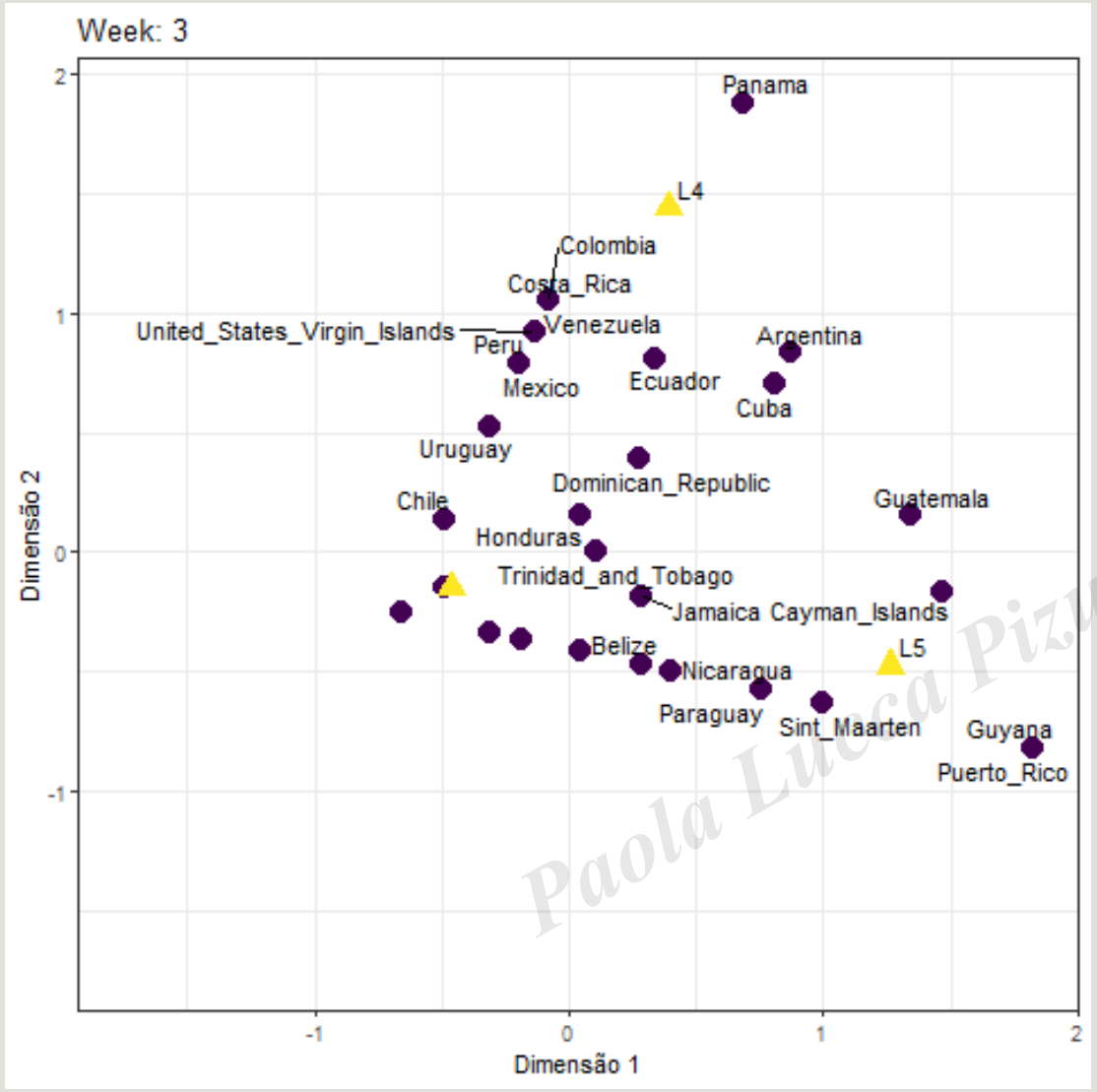




Análise Fatorial – PCA



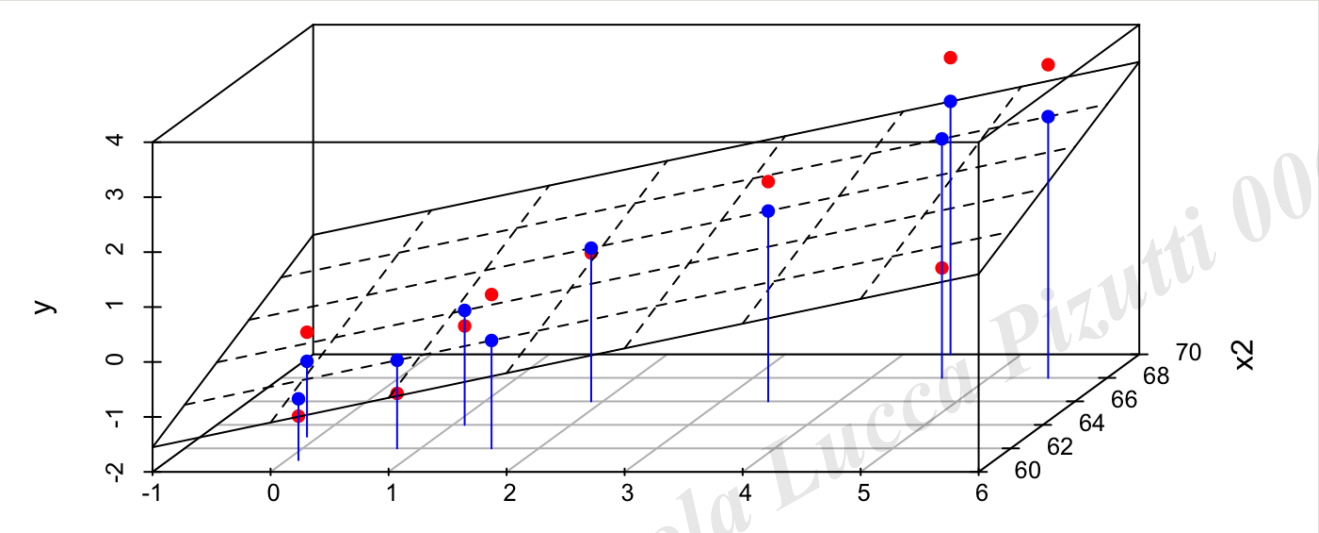




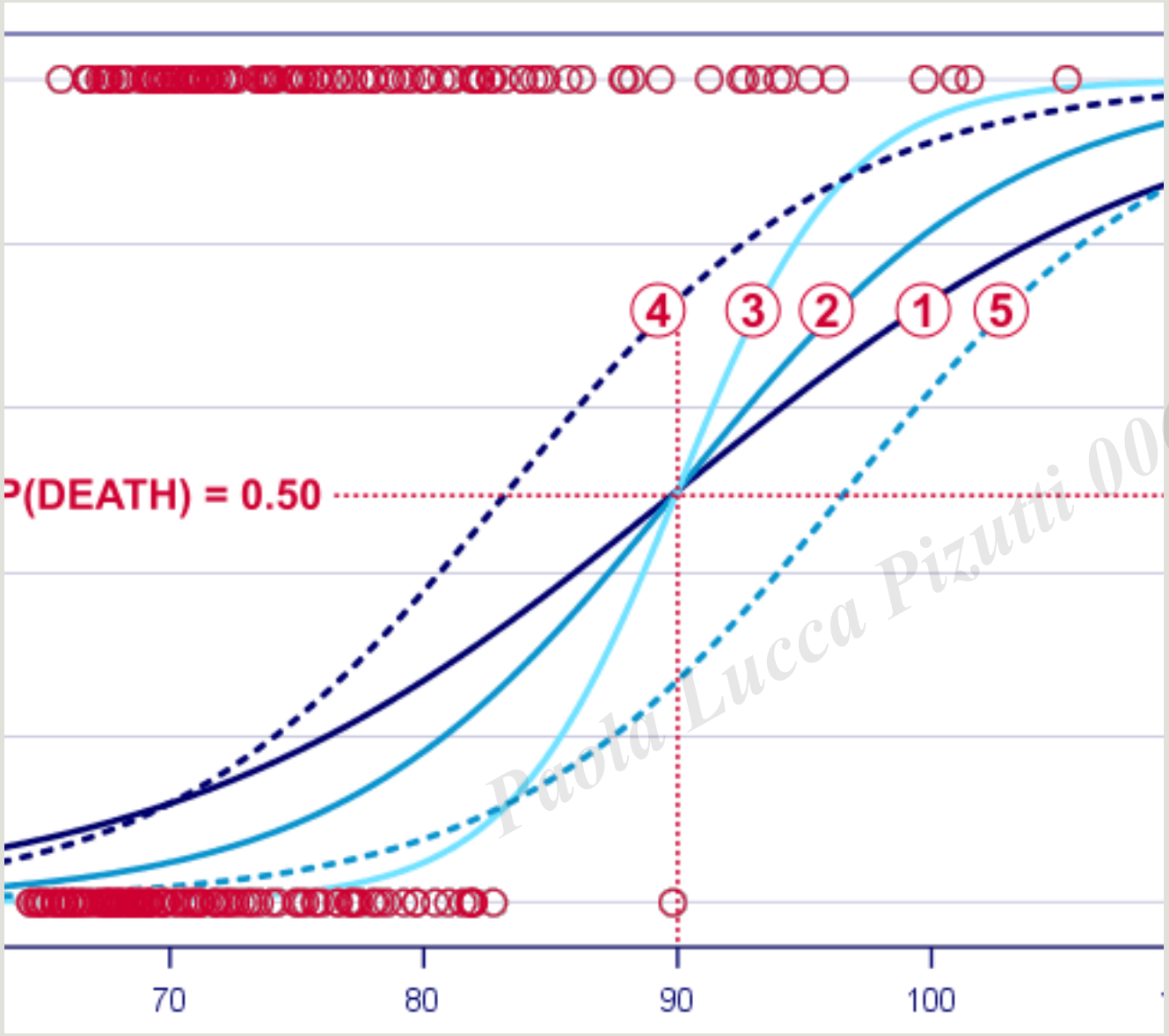
Análise de Correspondências Simples e Múltiplas

Algoritmos Supervisionados

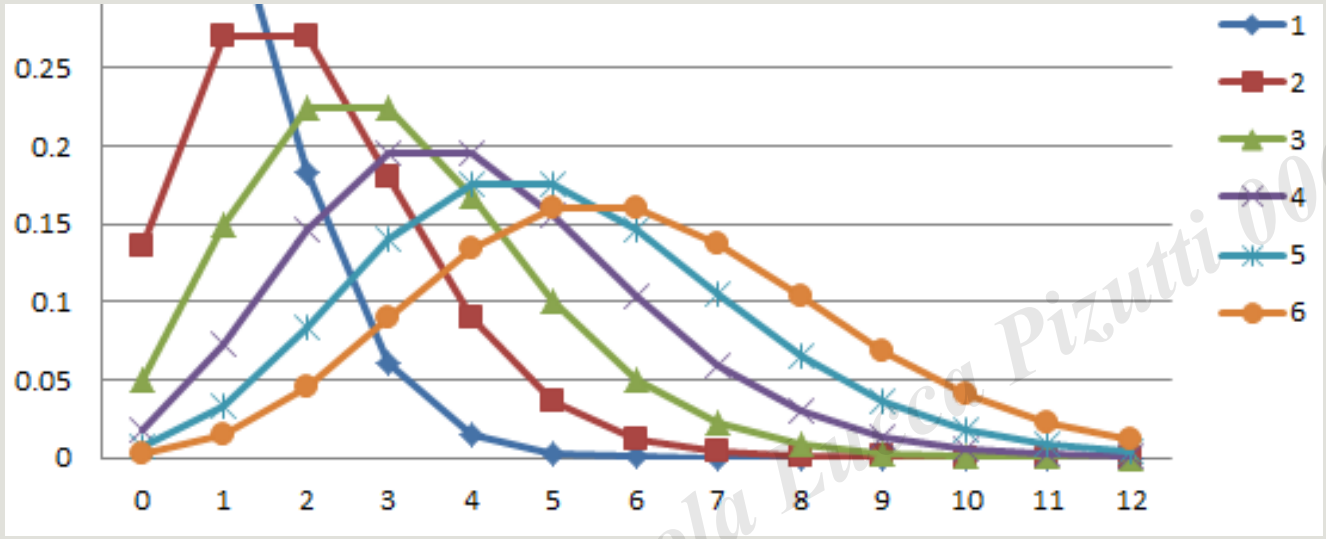
Generalized Linear Models	▪ Regressões Lineares Simples e Múltiplas; ▪ Regressões Logísticas Binárias e Multinomiais; ▪ Regressões para Dados de Contagem; ▪ Regressões para dados de Contagem com a Inflação de Zeros.
Generalized Linear Mixed Models	▪ Regressões Multinível Lineares de 2 e 3 Níveis.
Ensemble Models e *Neural Networks*	▪ Árvores de Decisão de Classificação ou de Regressão; ▪ *Boosting*; ▪ *Bagging*; ▪ *Random Forests*; ▪ Redes Neurais Artificiais.



Regressões Lineares

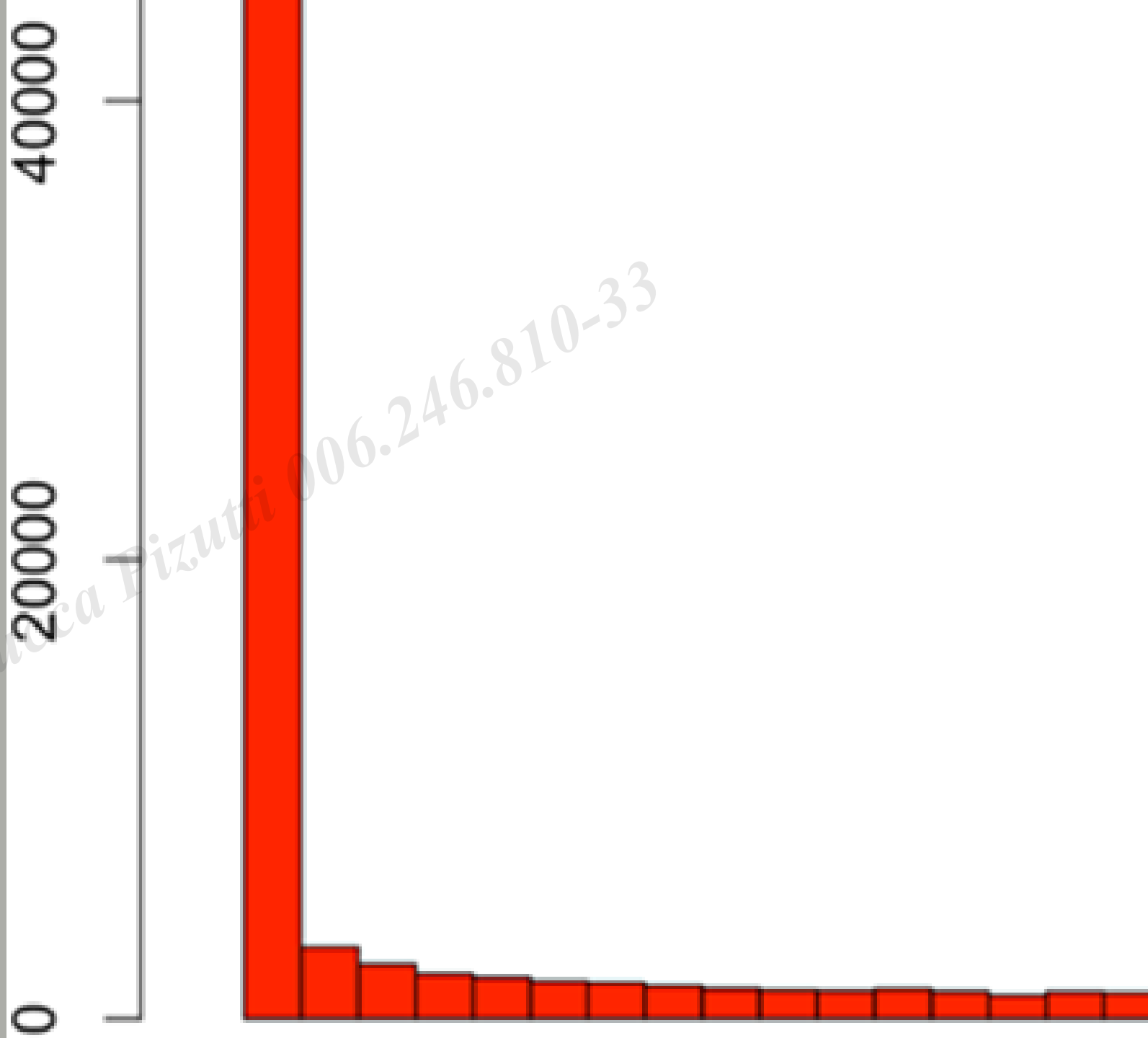


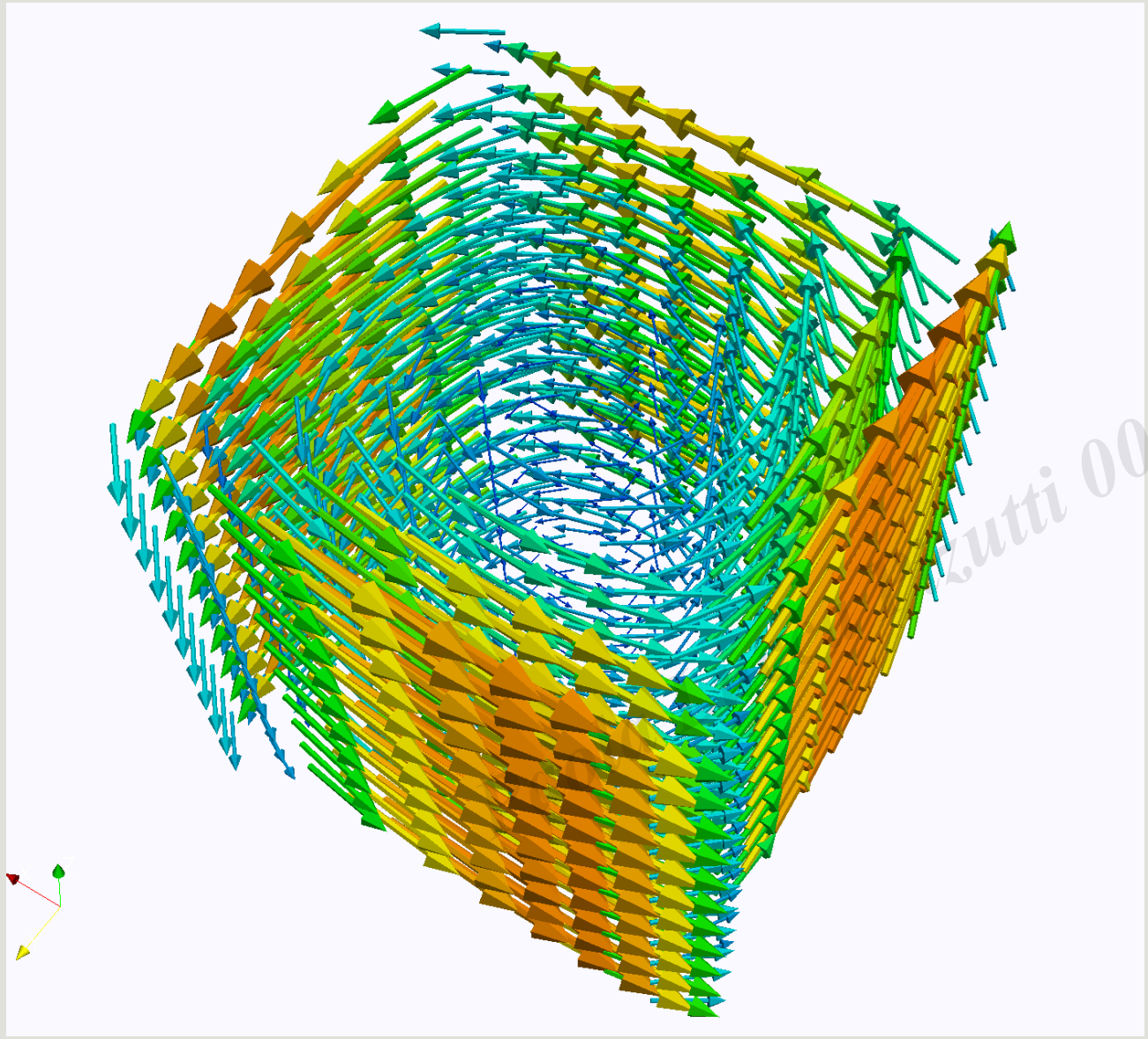
Regressões Logísticas



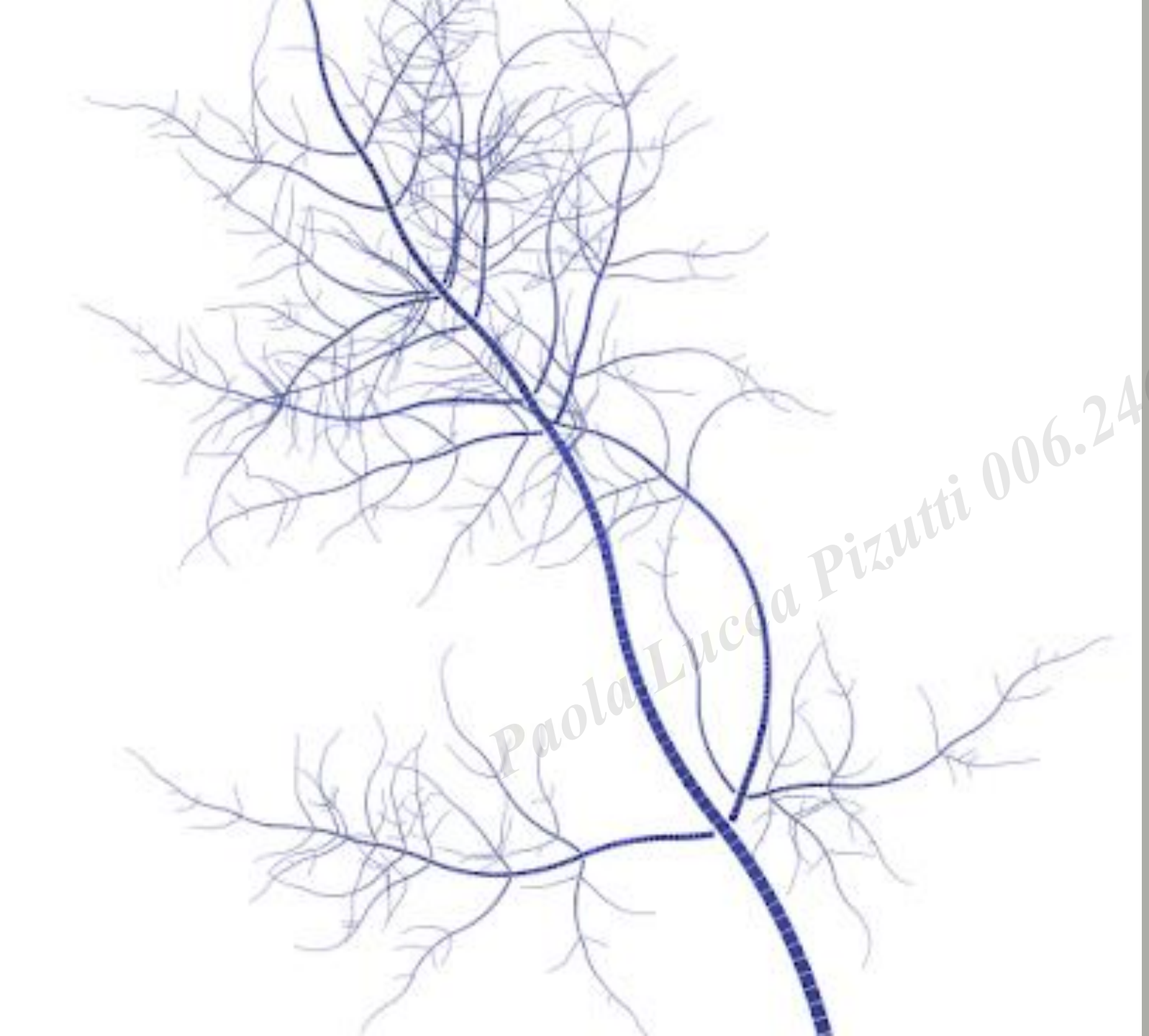
Regressões para Dados de Contagem

Regressões para Dados de Contagem com Zeros Inflacionados

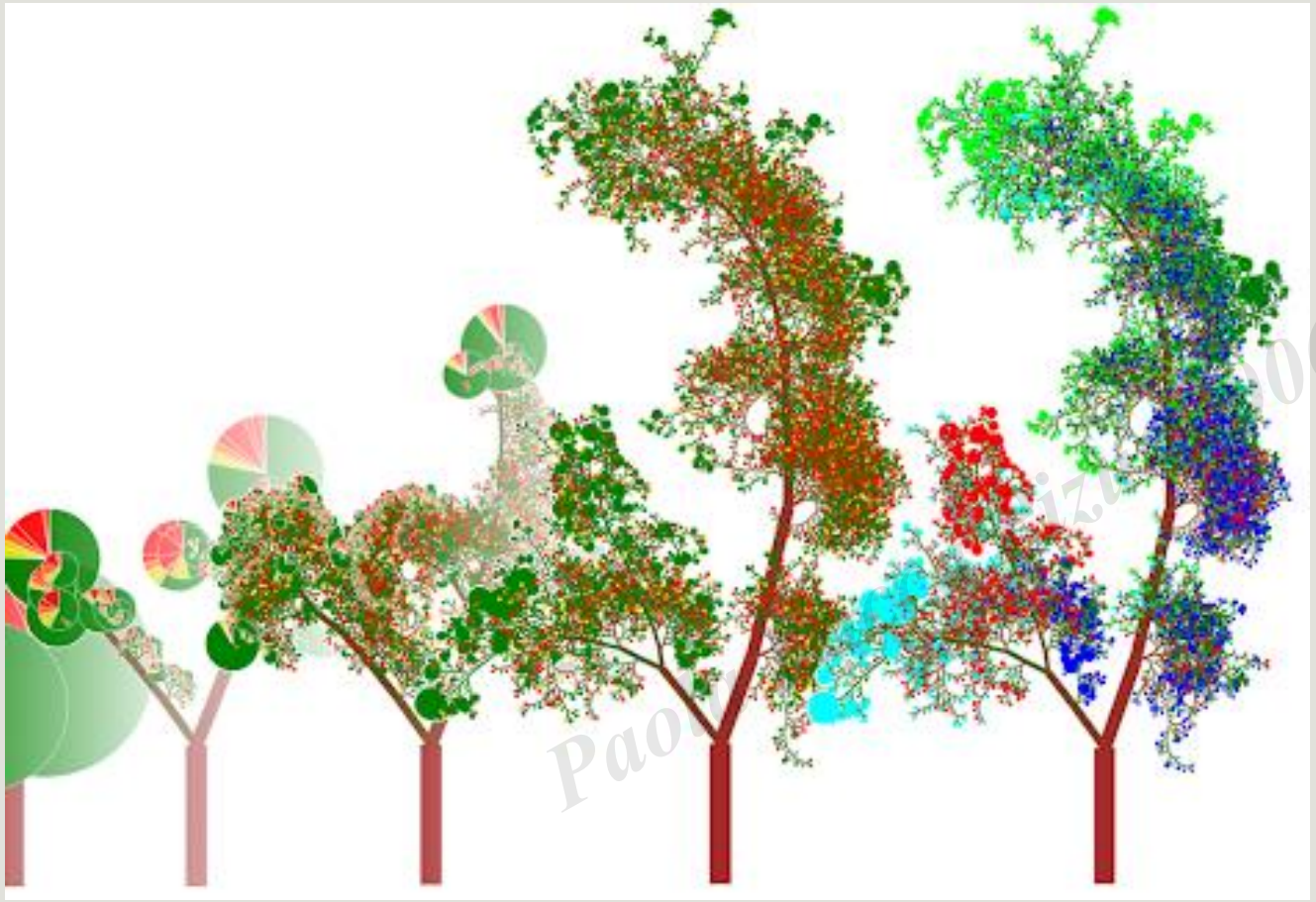




Regressões Multinível



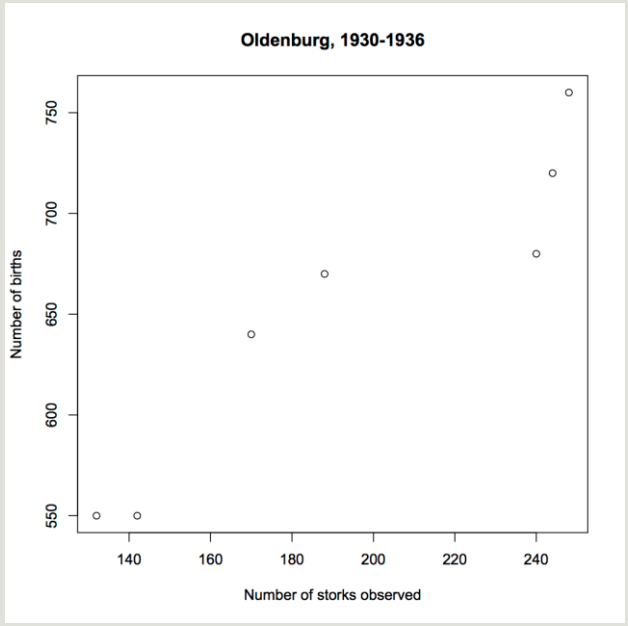
Árvores de Decisão



Random Forests

Redes Neurais

Relações de Causa e Efeito

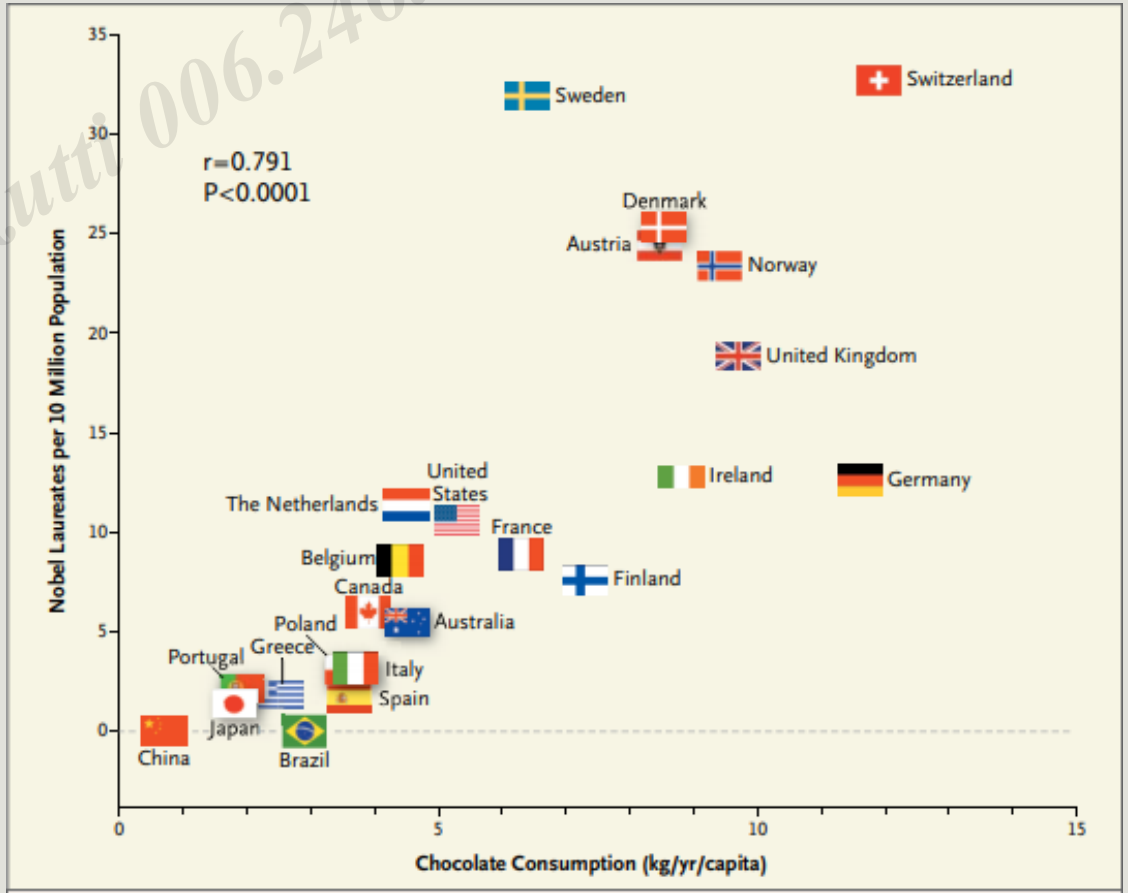


Exemplo Didático - 01

Dados adaptados por Box, Hunter, e Hunter de G. Fischer, Ornithologische Monatsberichte, vol. 44, no. 2, Jahrgang, 1936, Berlin e vol. 48, No. 1, Jahrgang, 1940, Berlin, e Statistiches Jahrbuch Deutscher Gemeinden, 27-33, Jahrgang, 1932-1938.

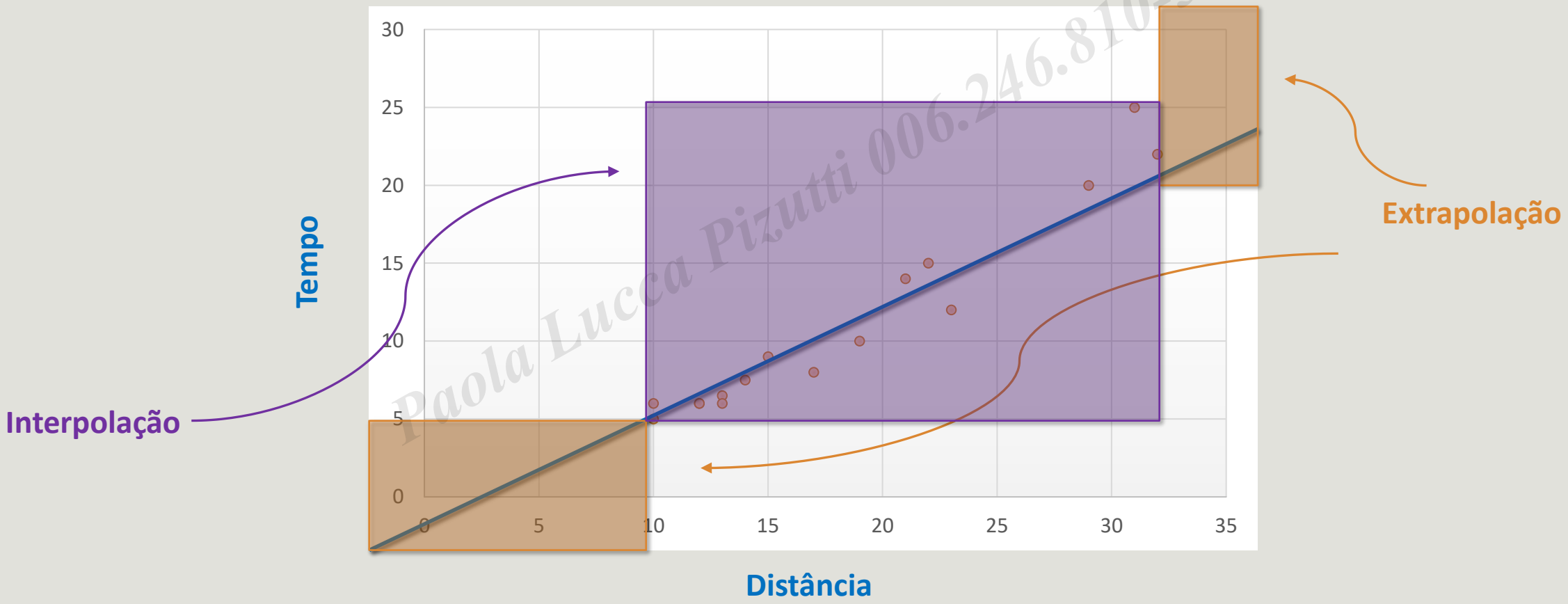
Exemplo Didático - 02

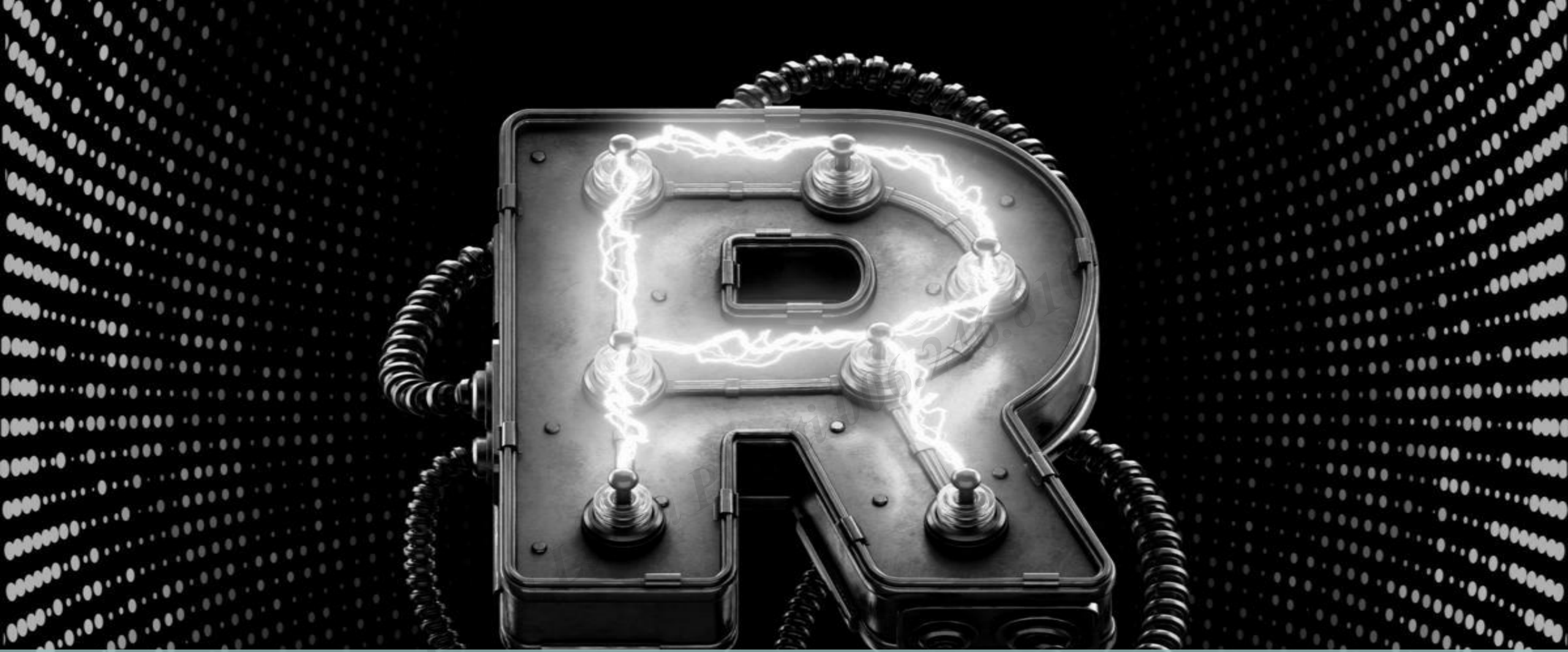
Dados de Messerli, F. (2012). Chocolate Consumption, Cognitive Function, and Nobel Laureates. The New England Journal of Medicine, 367, pp. 1562-1564



Interpolação x Extrapolação

Interpolação e Extrapolação





Introdução à Linguagem R

Por Que Aprender uma Linguagem de Programação?

- Aprender a programar é muito importante quando desejamos entender os dados! Mais: é importante para que possamos interpretar esses dados; para que possamos transformá-los em informação!
- Se você trabalha ou deseja trabalhar com dados, programar é uma skill de extrema relevância.

Uma pergunta justa seria:

"Eu já trabalho ou desejo trabalhar com dados, e já possuo acesso a eles via Windows, OSx, Android, iOS, Chrome, Mozilla, Edge, Safari... Ainda assim, eu preciso aprender a programar?"

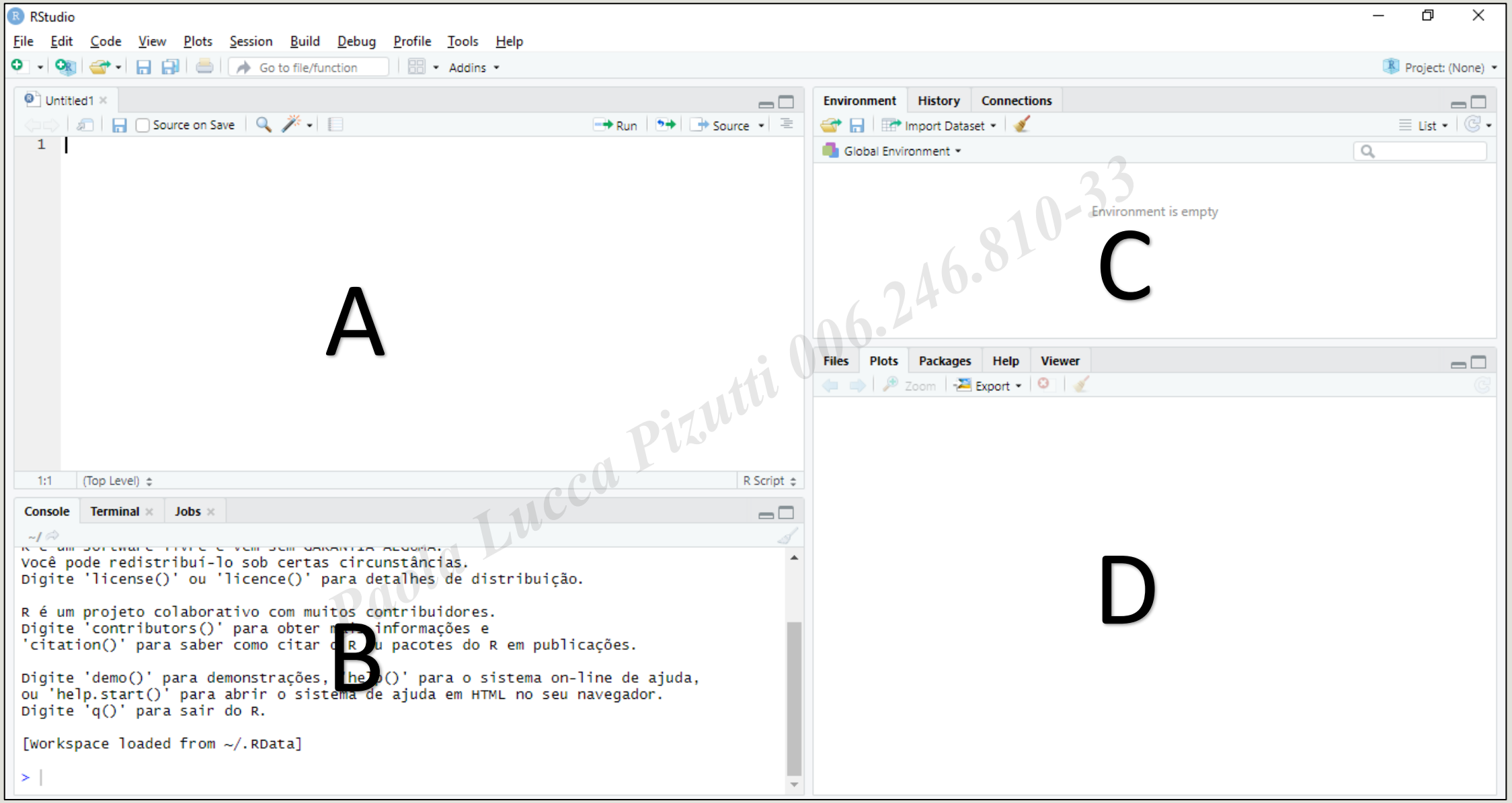
As Limitações de uma *Graphic User Interface* (GUI)

- Reprodutibilidade
- Automação
- Comunicação

Por Que o R?

Sobre o R

- A linguagem R surgiu em 1995, derivada da linguagem S, e é orientada a objetos.
- Possui inúmeros pacotes (mais de 17 mil), com vantagem para a aplicação da Estatística Avançada e uma vasta comunidade de suporte, além de fortes capacidades voltadas ao Data Science.
- Comprehensive R Archive Network (CRAN) é o repositório da linguagem R em que cada usuário pode contribuir com novos pacotes (coleções de funções em R com código compilado). Esses pacotes podem ser facilmente instalados com uma linha de código.
- Leituras recomendadas para quem não ainda não conhece a linguagem R e deseja se aprofundar:
  - Hands-On Programming with R (Grolemund, 2014);
  - R for Data Science (Wickham & Grolemund, 2016);
  - Ciência de Dados com R (Oliveira, Guerra & McDonnell, 2018).



Objetos, Funções e Argumentos

Há autores que definem os objetos do R como sendo uma variável. Para o curso, entenderemos que variáveis correspondem a características de uma dada amostra ou população.

- Objetos são maneiras simples de se acessar algo que foi salvo na memória da máquina. Pode ser um valor, uma palavra, uma ou mais variáveis, uma URL, uma base de dados amostral ou populacional, uma lista de coisas distintas contendo informações e tamanhos distintos, um gráfico, um mapa, uma imagem, um novo comando, etc. **No R TUDO é um objeto!** Cada um desses objetos possui uma classe!
- Funções correspondem a ações, a ordens direcionadas à máquina;
- Argumentos correspondem a um refinamento ou um melhor direcionamento das ações ou ordens propostas pelas funções.

Criando um Objeto no R

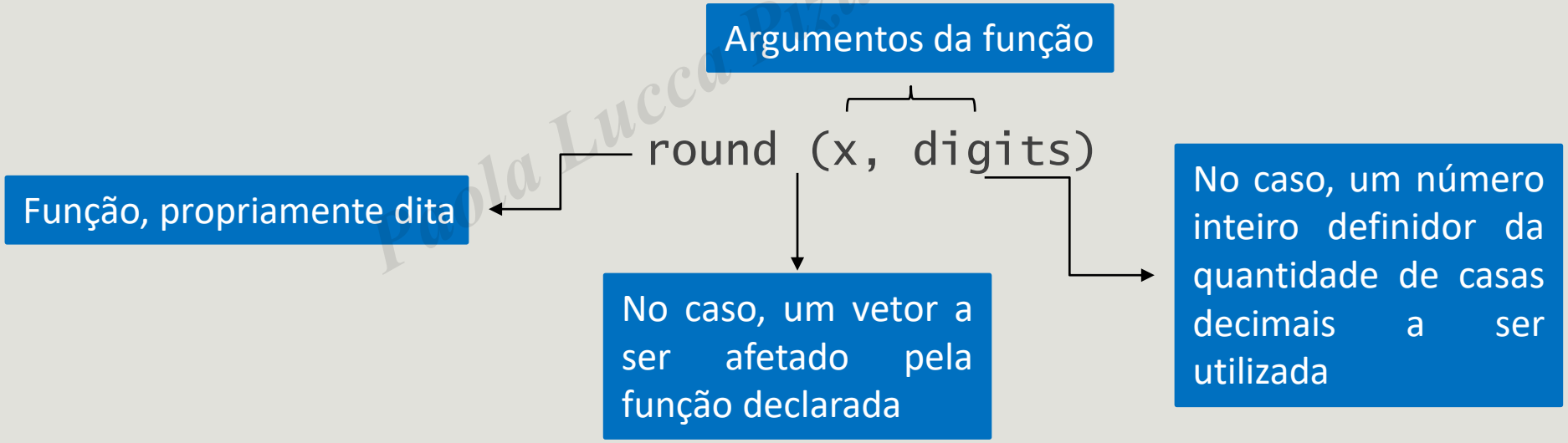
- Há duas formas de se criar um objeto na linguagem R:
  - Utilizando-se o símbolo =; ou
  - Utilizando-se o símbolo <− chamado operador de atribuição (prefira essa forma, reservando o símbolo = para as atribuições de valor dos argumentos das funções ou para operações matemáticas). O sinal de atribuição pode ser rapidamente declarado ao se pressionar conjuntamente as teclas 'Alt' e '-'.
- Os nomes dos objetos estabelecidos em R devem seguir certas regras:
  1. Não devem começar por números e nem por pontos;
  2. Não é desejável que contenham espaços, muito embora a situação seja possível com o uso dos símbolos ``;
  3. Suas nomenclaturas também não aceitam caracteres especiais, como: !, ~, $, @, +, -, /, *.
  4. Recomenda-se evitar nomear os objetos com os mesmos nomes de funções já estabelecidas;
  5. Recomenda-se não utilizar acentos e, sempre que possível, evitar as letras maiúsculas, visto que a linguagem é *case sensitive*.

Principais Funções Introdutórias do Curso

Função	Serve para:
args()	Verificar os argumentos de uma dada função no R
round()	Arredondar números
sample()	Criar amostras
class()	Verificar as classes dos objetos do R
View()	Visualizar objetos em forma de planilha
head()	Visualizar as primeiras observações de uma base de dados
tail()	Visualizar as últimas observações de uma base de dados
str()	Observar a estrutura de uma base de dados
length()	Observar o comprimento de um vetor ou de uma lista de dados
dim()	Descobrir as dimensões de um objeto
nrow()	Contar o número de linhas de uma base de dados
ncol()	Contar o número de colunas de uma base de dados
rm()	Remover um objeto do ambiente de trabalho
install.packages()	Instalar pacotes
library()	Carregar pacotes

Utilizando funções no R

Para utilizar uma função no R, devemos conhecer sua forma funcional, isto é, devemos, em regra, declarar os argumentos inerentes a ela. Exemplo de utilização da função `round()`:



Pacotes na Linguagem R

- A linguagem R possui milhares de pacotes direcionados às mais diversas áreas do conhecimento, e a maioria não está instalada em nossos computadores. Para instalar um pacote, devemos comandar:

```
install.packages("nome do pacote aqui")
```


- A instalação de um pacote não basta para sua utilização. Assim, a cada nova seção aberta do RStudio, devemos chamá-los da seguinte maneira:

```
library(nome do pacote aqui)
```

Criando e Excluindo Variáveis em um *Dataset*

- Criando uma variável:

```
basededados$nova_variável <- NA
```

ou

```
mtcars["nova_variável"] <- NA
```

- Excluindo uma variável:

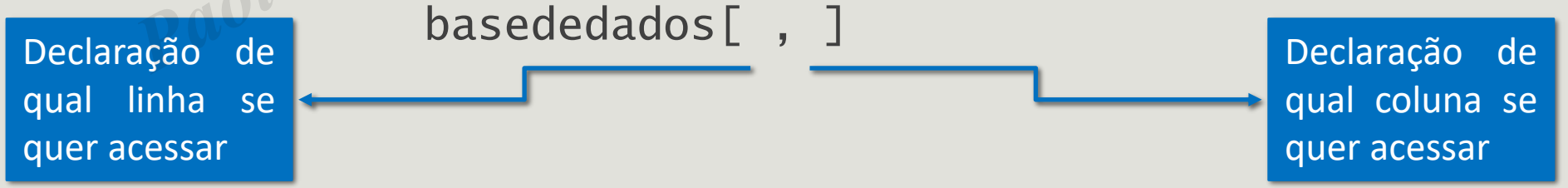
```
basededados$nova_variável <- NULL
```

Extraindo Valores de um *Dataset*

- Para extrair uma coluna de um *dataset*, utilize o operador `$`:

```
basededados$nova_variável
```

- De maneira mais precisa, também se pode extrair valores do *dataset* com o operador `[ , ]`:



Funções `if`, `else` e `ifelse`:

```
if(teste lógico){
        caso a resposta do teste lógico seja TRUE, faça isso
} else {
        caso a resposta do teste lógico seja FALSE, faça essa outra coisa
}
```

```
ifelse(teste lógico,
     yes = caso a resposta do teste lógico seja TRUE, faça isso,
     no = caso a resposta do teste lógico seja FALSE, faça essa outra coisa)
```

Funções `for`, `while` e `repeat`

```
y <- 10

for (i in 1:5){
      print(y + i)
}
```

Para cada `i` (poderia ser qualquer símbolo ou palavra), presente na sequência de 1 a 5, imprima o valor da soma entre `y` e `i`

Funções `for`, `while` e `repeat`

```
z <- 0

while(z < 10){
        print(z)
        z <- z + 1
}
```


Enquanto z for menor do que 10, imprima z e, depois, atualize o valor de z acrescendo o seu valor em uma unidade

Funções `for`, `while` e `repeat`

```
w <- 3


repeat{
        print(w)
        w <- w + 2
        if(w > 18) break()
}
```


Repita os passos a seguir:

- Imprima o valor de w;
- Atualize o valor de w, acrescendo-o em duas unidades;
- Se w passar a ser maior do que 18, pare tudo.

Visualização de Dados com o ggplot2

- A sintaxe mais básica do pacote ggplot2, para a criação de um gráfico a partir de um *data frame*, é a seguinte:

```
ggplot(data = base de dados aqui) +
   geom_geometria escolhida aqui(aes(principais elementos do gráfico aqui))
```

Rafael de Freitas Souza
Linkedin