

MBA⁺

**ARTIFICIAL INTELLIGENCE
& MACHINE LEARNING**

MBA⁺

PROGRAMANDO IA COM R

Prof. Elthon Manhas de Freitas
elthon@usp.br

2018

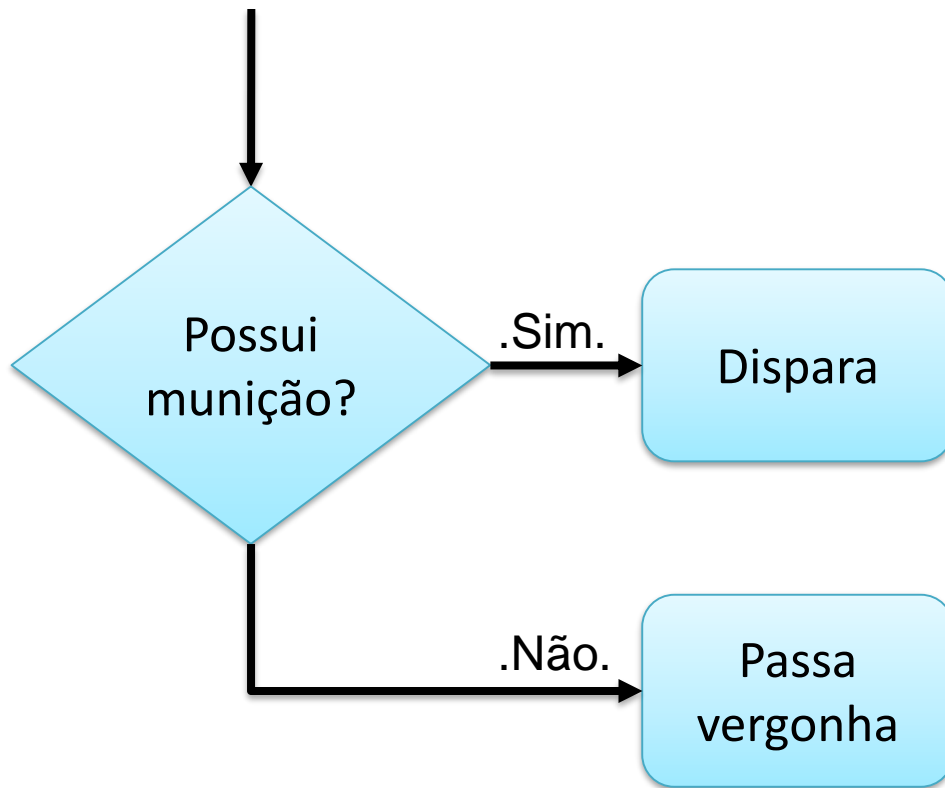
Revisão da última aula

- O que vimos na aula passada?



Controle do Script – Basic Scripting

Condicionais



- IF

```
if (condição) {  
    //Instruções  
}
```

- IF / ELSE

```
if (condição) {  
    //Instruções  
} else {  
    //Instruções  
}
```

- IF / ELSE IF / ELSE

```
if (condição 1) {  
    //Comandos  
} else if (condição 2) {  
    //Comandos  
} else {  
    //Comandos  
}
```

Condicionais – Ternários

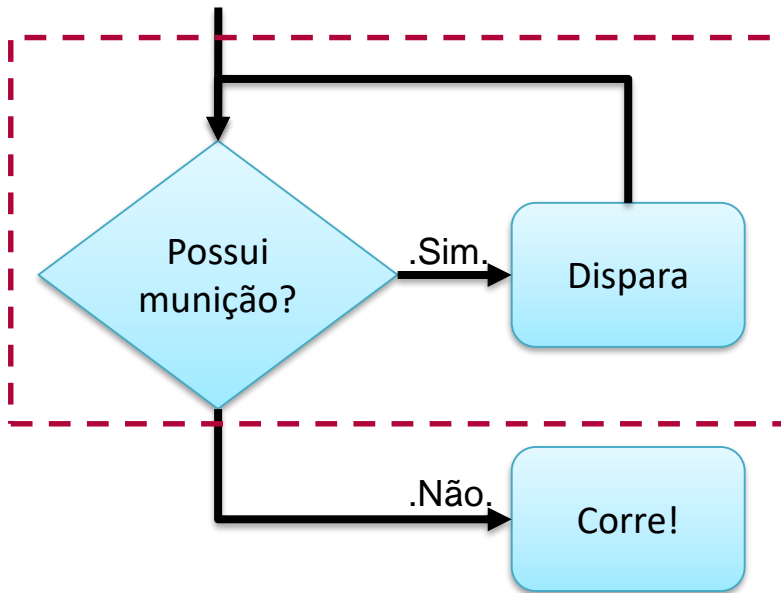
```
variável <- ifelse(condição, valor V, valor F);
```

```
if (condição){  
    variável = (valor V);  
} else {  
    variável = (valor F);  
}
```



Loop – Laços

- Laços são usados em programação quando precisamos que um mesmo trecho de código aconteça diversas vezes.



- Muito útil quando se sabe previamente a quantidade de “iterações” que o laço deve executar.
- Exemplo de `for`

```
for (i in vetor) {  
    //Comandos  
}
```

Tipos de vetores? E Matrizes? E tabelas?

Também podemos usar seqüências

- `seq`
- `seq_along`
- `etc.`

- A instrução while indica que tudo o que estiver dentro do laço, será executado enquanto a condição do laço for verdadeira.

```
while(condição){  
    //Comandos  
}
```

- Perceba que não há declaração de variável no while, nem parte de atualização / incremento.

Loop – WHILE

- Mesmo código, escrito com while

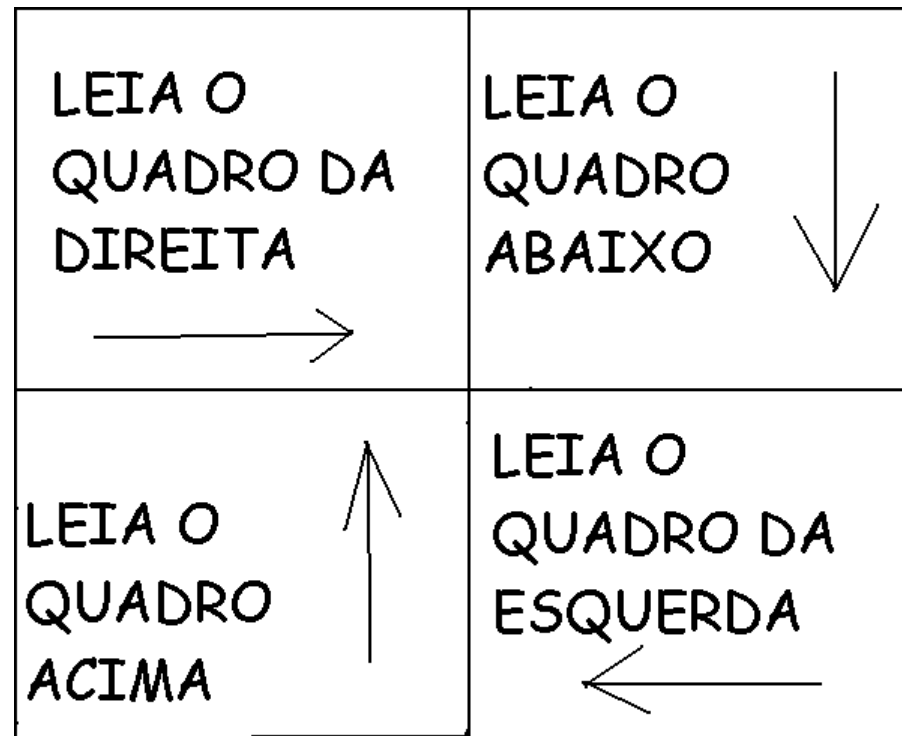
```
strength = 1000.0;
for (i in 1:12){
    strength = strength * 1.01;
}
print( strength )
```

```
strength = 1000.0;
i = 1;
while(i <= 12){
    strength = strength * 1.01;
    i = i + 1
}
print( strength )
```

Loop – BREAK / CONTINUE

- Utilizados para alterar o fluxo do loop
- `break`
 - Utilizado para “quebrar o laço”, ou seja, sair do loop a qualquer custo.
- `next`
 - Utilizado para ir para a próxima iteração do loop, sem executar os próximos comandos dentro do loop.

Loop infinito



Loop infinito

FIAP



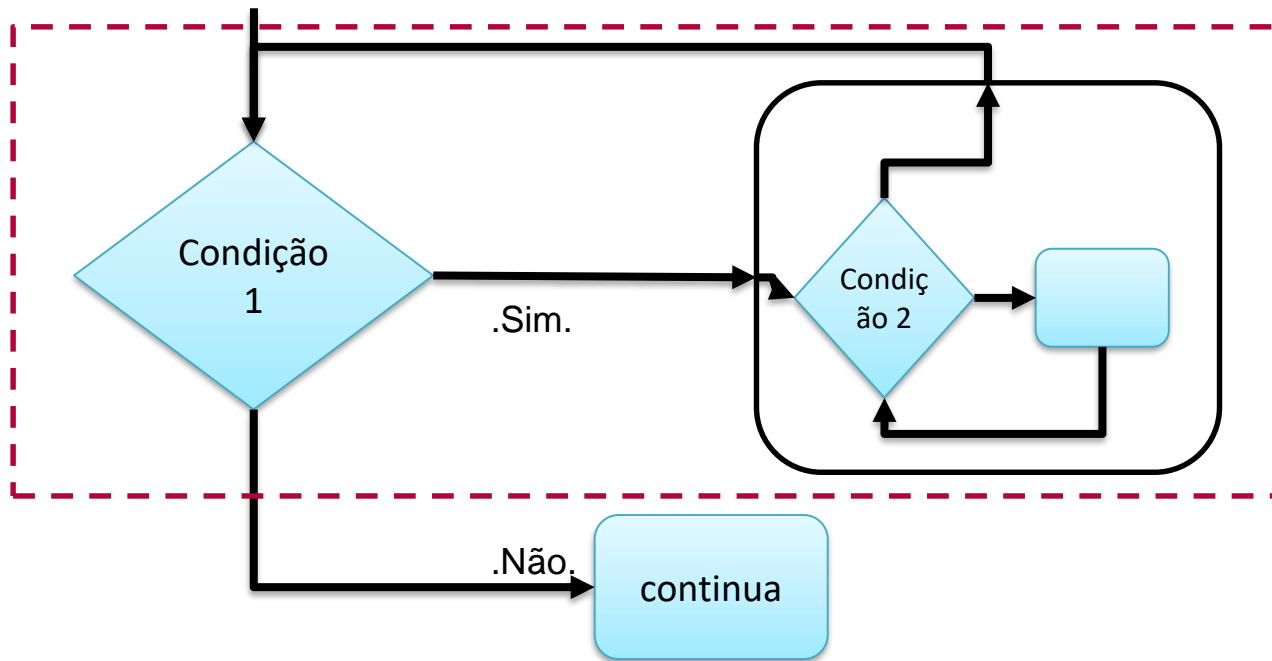
Loop – REPEAT

- Comando usado para causar um loop infinito
 - O laço só para com um `break`
 - (ou com `return`, no caso de uma função, que veremos em seguida)

```
x0 <- 1
tolerancia <- 1e-8

repeat {
  x1 <- computeEstimate(x0)           #Função fictícia
  if(abs(x1 - x0) < tolerancia) { ## Suficiente?
    break
  } else {
    x0 <- x1
  }
}
```

Loops encadeados – nested loops



Exercício – Loop Acumulado

- Criar script que varre o dataset `AirPassengers`
 - Observe que este dataset não é matrix, table ou data.frame, mas um TimeSeries
- Crie um vetor que contenha o valor acumulado de passageiros ao longo do tempo.

Mês	Passageiros	Esperado
1	5	5
2	4	9
3	10	19
...
n	12	$\sum_{i=1}^n passageiros[i]$

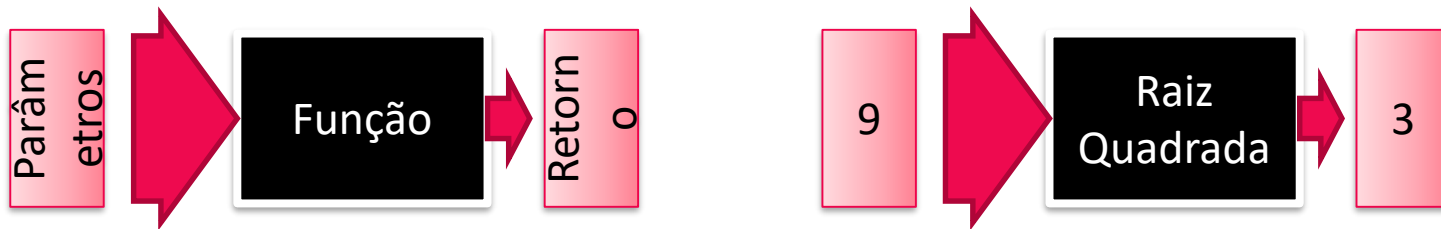
```
Resposta FIAP
# acum <- c()
# for (i in AirPassengers) {
#   acum <- c(acum, total(acum,i) + i)
# }
# acum <- acum[1:12]
# acum
```



Métodos: Funções e Sub-rotinas

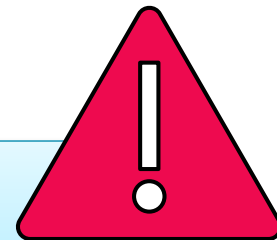
- Trata-se de um conjunto de instruções “empacotadas” para serem reutilizadas (sub-algoritmo);
- Podem ser vistas como um conjunto de procedimentos agrupados para um determinado fim;
- Podem receber parâmetros;
- Interessante enxergá-las sob o conceito “atômico”
 - Que não pode ser cortado, indivisível.

- Similares às sub-rotinas, porém retornam valores.
 - Obs.: Em R, as sub-rotinas são funções que não tem retorno



O retorno de uma função pode ser dado:

- Pela instrução “return(valor)”
- Pelo último comando executado na função!



Sintaxe de uma função

- Para declarar a função:

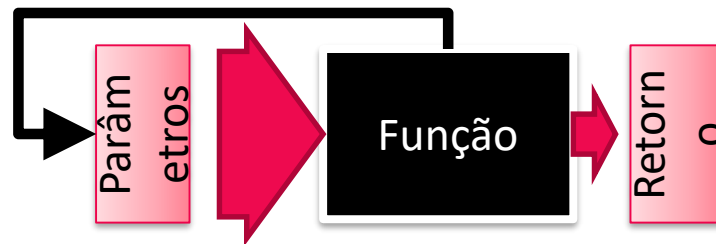
```
obj_name <- function(<parâmetros>) {  
    instruções  
}
```

- Para executar a função:

```
obj_name(<parâmetros>)
```

Funções Recursivas (overview)

- São funções que dentro delas, chamam a si mesmas;
- Propensas a Loop infinito;



- Criar função Fatorial que retorna o fatorial de um número inteiro.
 - O fatorial de um número n é $n * n-1 * n-2 \dots$ até $n = 1$
 - Exemplo $4!$ (lê-se 4 fatorial) é dado por:
 - $4 * 3 * 2 * 1$
- Fazer uma função que recebe um vetor numérico e retorna seus valores invertidos.
 - Exemplo:
 - Entrada: 44, 67 , 5, 47, 8, 1, 79, 128
 - Saída: 128, 79, 1, 8, 47, 5, 67, 44

- Fazer uma função que dá a mesma resposta do Exercício “Loop Acumulado”
- Passar como parâmetro a Série `AirPassengers`
- Passar como parâmetro a coluna `mpg` da tabela `mtcars`
 - Perceberam que pode ser acessada por `mtcars$mpg` ?

Exercício – Loop Acumulado

FIAP

- Criar script que varre o dataset `AirPassengers`
 - Observe que este dataset não é `matrix`, `table` ou `data.frame`, mas um `TimeSeries`
- Crie um vetor que contenha o valor acumulado de passageiros ao longo do tempo.

Mês	Passageiros	Esperado
1	5	5
2	4	9
3	10	19
...
n	12	$\sum_{i=1}^n \text{passageiros}[i]$



Para raciocinar ...

- O que é a variável `i`? Uma função ou um número?
- O que é esperado da “Execução 1”?
- E da “Execução 2”?

```
h <- function() {  
  x <- 10  
  function() {  
    x  
  }  
}  
i <- h()
```

```
#Execução 1:
```

```
i()
```

```
x <- 20
```

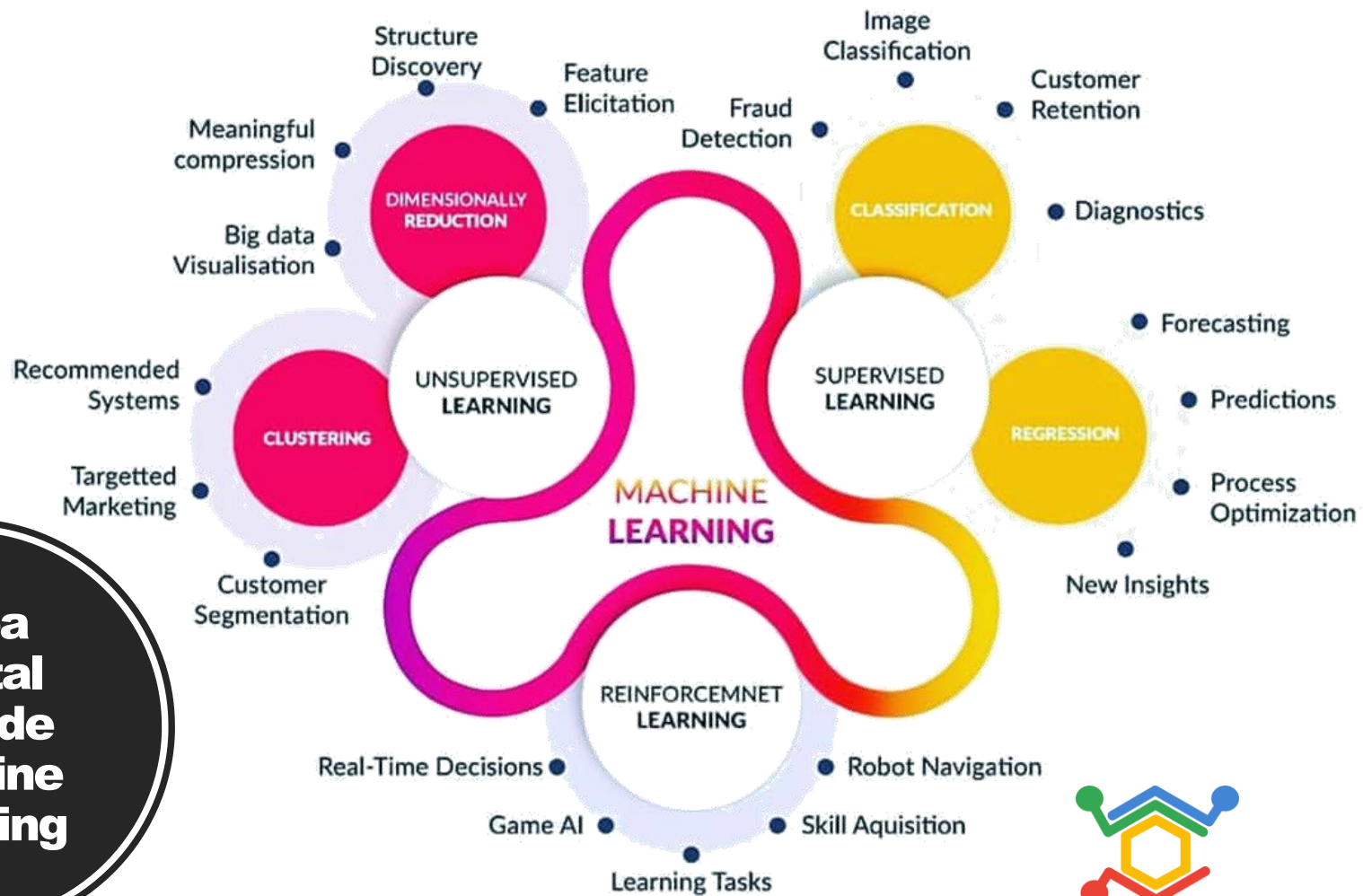
```
#Execução 2:
```

```
i()
```



Análise de dados – Motivação

Mapa Mental usos de Machine Learning

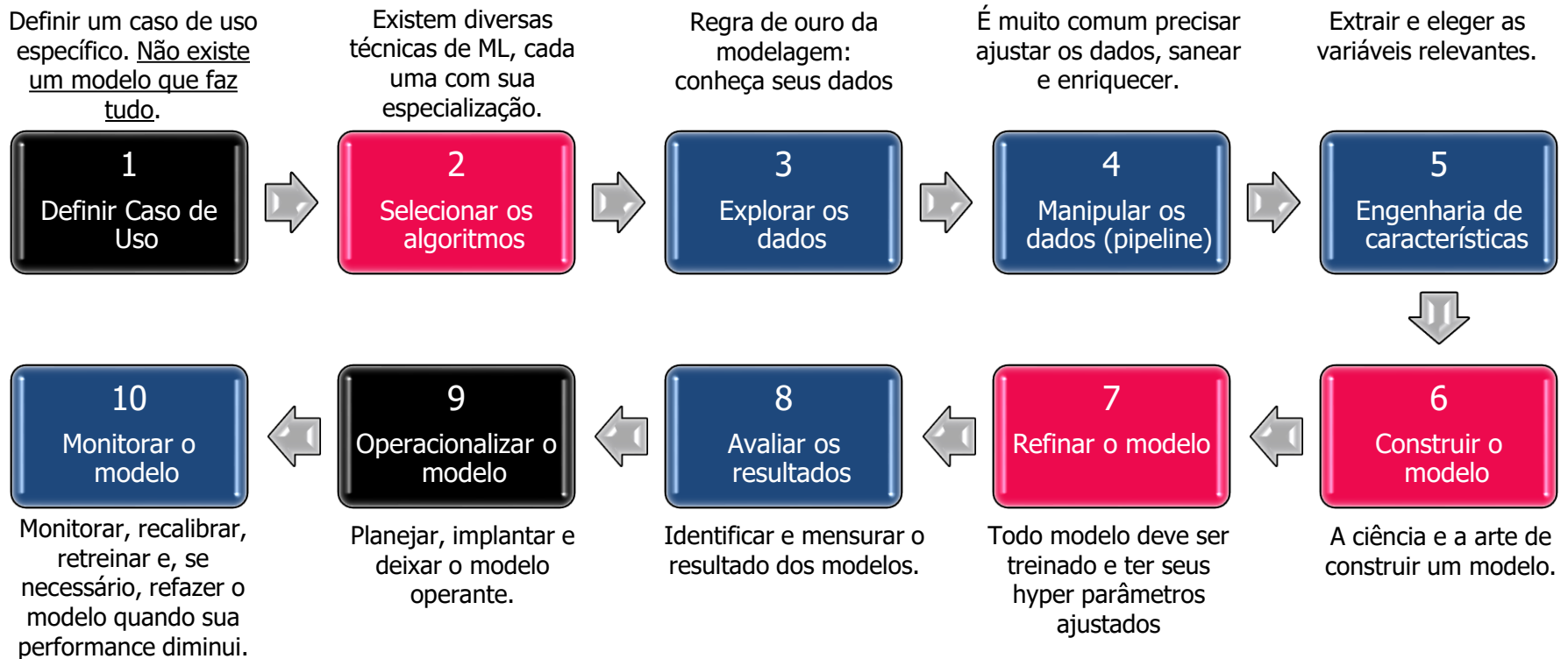


Dopamine
ago-2018

<https://github.com/google/dopamine>

Machine Learning Project Lifecycle

- Detalhes das etapas e principais skills requeridos

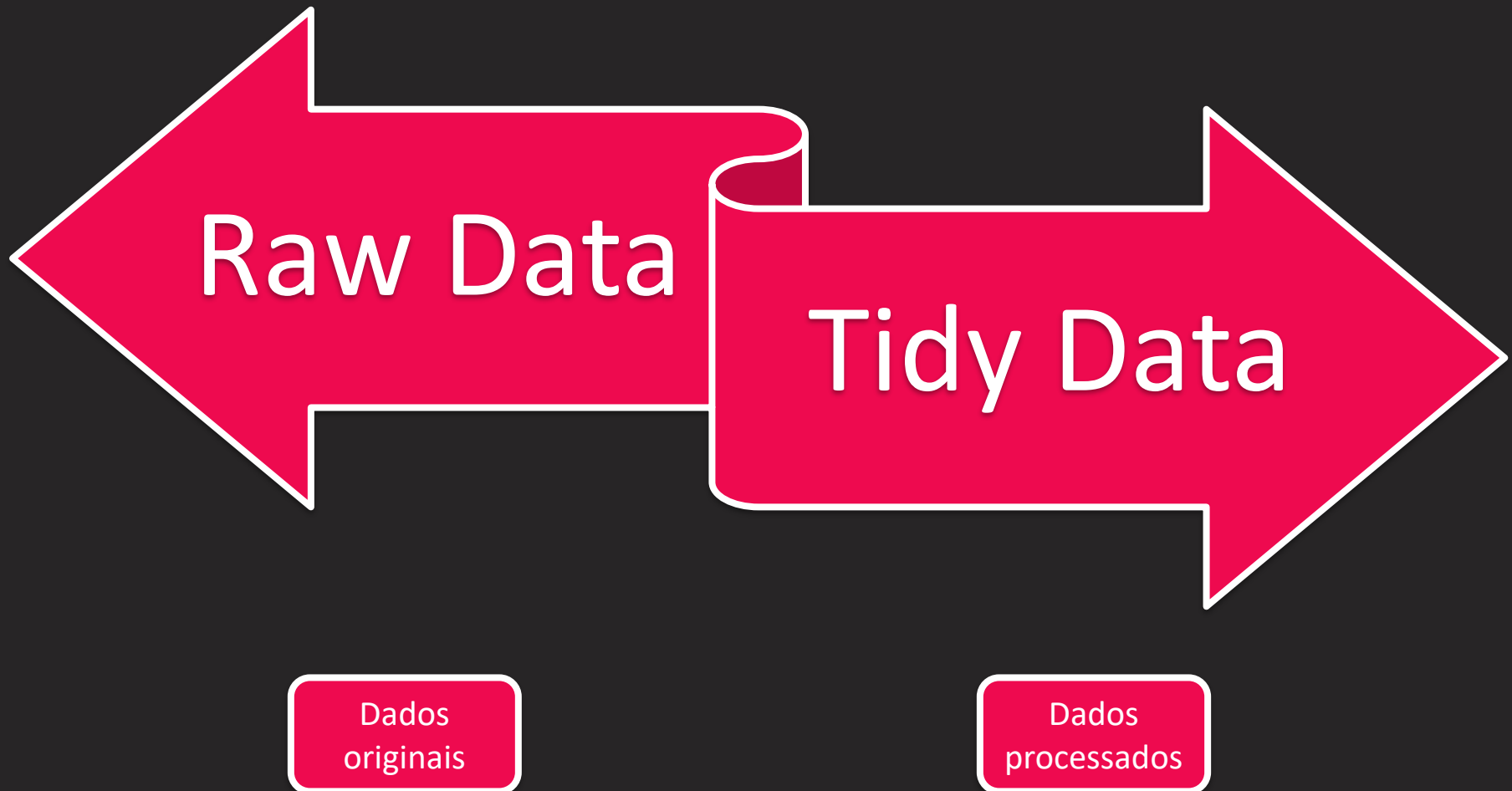


Legenda - Skill Principal

Projeto

Machine Learning

Analytics



- O que são os dados originais?
 - São os dados obtidos da origem sem nenhuma manipulação humana ou sistêmica (exceto operacionais)
 - Geralmente são difíceis de serem analisados
 - Necessita de um processamento inicial
 - No ciclo da análise e modelagem, devem ser processados apenas UMA VEZ!
 - Se necessário reprocessamento, o ideal é testar ou reiniciar etapas posteriores

Raw data – Dados Originais

- Os dados originais incluem:
 - Aqueles binários estranhos que a máquina ‘cospe’ em alguns processos
 - O monte de excel sem formatação que as áreas usam para diversos fins
 - Aquele JSON complicado que o scrapper gerou
 - Números anotados a mão nos caderninhos dos vendedores em campo

- O que são os dados tratados?
 - São os dados processados, prontos para análise
 - (análise humana ou de máquina)
 - Processar inclui: Merge, Subset, Transformação, Enriquecimento, etc.
 - Tem que haver padrão para o processamento
 - Todos os passos devem ser gravados



Data Wizard

Você até pode dizer
que tem outra
profissão



Continua ... |



Exercícios individuais

- Aprenda R no R**
- Portfólio individual**

lembrete: 2 pontos cada

MBA⁺

Copyright © **2018**

Prof. Elthon Manhas de Freitas

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).