

PROGRAMAÇÃO EM R

AULA 04 – Criando um projeto completo no R

AGENDA

- Estruturas de controle:
 - Condicionais: If-Else;
 - Repetição: Loop For; Loop While ; Rep.
- Funções;
- Criando um projeto;
- Resumo curso;
- Projeto completo.

ESTRUTURAS DE CONTROLES

- Objetivos:
 - Fazer validações;
 - Repetir uma parte do código;
 - Fazer mudanças no comportamento do script, a partir de definições prévias.

ESTRUTURAS DE CONTROLES

- If-Else
 - If(condição) {tarefa1} else {tarefa 2};
 - Ifelse(condição, tarefa1, tarefa2);
- Loop For
 - For(i in 1:N) {tarefa};
- Loop While
 - While (condição ok) {tarefa};
- Repeat
 - Repeat{statement};
- Outras instruções de controle: Break e Next.

FUNÇÕES

- É um bloco de códigos;
- Objetivos:
 - Evitar repetição de código;
 - Permitir que o código fique mais legível e elegante;
 - Forma de modularizar o código;
- Sintaxe de uma função:
 - Nome_funcao(parâmetros)
 {código a ser executado}
- Os parâmetros de uma função possuem um escopo local, isto é, não afetam o restante do código do script;
- Funções built-in:
 - São funções interna do R;
 - Exemplos: sum(), mean(), round(), etc.

FUNÇÕES

■ Exemplos funções:

```
CalculaMedia <- function(P1, P2, TR, LT){  
  # Primeiro foi feita a validação dos dados  
  if(is.na(P1)){  
    P1 = 0}  
  if(is.na(TR)){  
    P2 = 0}  
  
  if(is.na(P1)){  
    TR = 0}  
  if(is.na(P2)){  
    LT = 0}  
  media=(0.3*P1) + (0.4*P2) + (0.2*TR) + (0.1*LT)  
  
  return(media)  
}
```

FUNÇÕES

■ Exemplos funções:

```
# Função Raízes Quadráticas
RaízesQuadrática <- function(A, B, C){

  # Total de linhas nos vetores
  N = length(A)

  # ToDo: FAZER VALIDACAO DO TAMANHO A=B=C
  if(N != length(B)){
    stop("Erro: Vetores de tamanhos diferentes")
  }

  if(N != length(C)){
    stop("Erro: Vetores de tamanhos diferentes")
  }

  # print("Tamanho do vetor")
  print(N)

  # Variável de retorno da função
  ret <- matrix(NA, ncol = 2, nrow = N)
  # Pegado os valores dos coeficientes
  for (linha in 1:N) {
    a <- A[linha]
    b <- B[linha]
    c <- C[linha]

    if(a == 0){
      print("Warning: 'a' tem que ser diferente de zero")
      next()
    }
  }
}
```

■ Continuação:

```
# Calcula o delta
Delta = b^2 - 4*a*c

# print(Delta)

# Avalia o delta e calcula as respectivas raizes
if(Delta > 0){
  # Se Delta maior do que zero

  x1 = (-b + sqrt(Delta))/(2*a)
  x2 = (-b - sqrt(Delta))/(2*a)

  raiz_1 = min(x1, x2)
  raiz_2 = max(x1, x2)

} else if (Delta == 0) {
  # Se Delta for igual zero

  raiz_1 = -b/(2*a)
  raiz_2 = NA

} else if (Delta < 0) {
  # Se Delta for menor do que zero
  # NESTE CASO NAO TEMOS RAIZ REAL
  raiz_1 = NA
  raiz_2 = NA
}

# Coloco o resultado das contas na matriz de retorno
ret[linha, 1] <- raiz_1
ret[linha, 2] <- raiz_2

} # FIM DO FOR (INTERACAO POR LINHA)
```


CRIANDO UM PROJETO

- Um projeto é uma forma de organizar melhor o conjunto de arquivos que serão utilizados em uma análise de dados;
- Podemos criar um projeto pelo menu: File- New Project;
- Podemos criar o projeto com o pacote ProjectTemplate;
 - Cria um esqueleto de pastas para guardarmos as configurações, dados, scripts, etc de forma estruturada;
 - Função `create.Project()`: criar projeto;
 - Função `load.project()`: carregar projeto (diretório tem que ser setado para projeto carregar);
- Podemos carregar uma função criada ou outro script do R;
 - Função `source()`;
- Debug;
 - Breakpoints;
 - Função `traceback()`: representa a sequência de funções executadas que geraram um determinado erro.

■ Fundamentos do R

- Tipos de operadores;
- Tipos de variáveis;
- Tipos de dados;
- Tipos de estrutura de dados;
- Operações com as estrutura de dados;

■ Carga, limpeza, transformação e manipulação

- Carga de arquivos .txt, .csv e planilhas excel;
- Pacote dplyr e suas funções: select, mutate, filter, arrange, summarize, group_by, join, pivot;
- Pacote tydr e suas funções: gather, spread, separate e unite;
- Reshape2;
- Subconjuntos;

■ Gráficos

- Gramática do dos gráficos: a construção dos gráficos em camadas;
- Pacotes de gráficos nativos do R: graphics e grdevices;
- Ggplot2;
- Principais tipos de gráficos: dispersão, linha, barras, boxplot, histograma, pizza.

■ Estruturas de controle: e funções

- Condicionais: If-Else;
- Repetição: Loop For; Loop While ; Rep.
- Funções.

■ Criando um projeto.

1. Criando um projeto no R

Crie um projeto no RStudio com a seguinte estrutura:

- Um diretório chamado: “Database”.
- Um diretório chamado: “Graficos”.
- Baixe a base de dados do site <https://ourworldindata.org/coronavirus-source-data>.
- Salve a base no diretório “Database” com o nome “owid-covid-data.xlsx”.

Caso você não esteja utilizando o RStudio apenas organize seu diretório de maneira a seguir a estrutura sugerida.

2. Camada de dados(Data Loader)

Crie um script chamado “DataLoader.r”, o script deve realizar as seguintes tarefas:

- Abrir o banco de dados de nome “owid-covid-data.xlsx” que está no diretório “Database”.
- A coluna data deve ser convertida para o tipo data. (garantindo que a coluna é do tipo “Date”).
- O script deve filtrar a base e mostrar apenas os seguintes países: “Brazil”, “United States”, “Mexico”, “Germany”, “France”, “United Kingdom”.
- O script deve selecionar apenas as variáveis: location, date, total cases, new cases.

3. Função

- Crie uma função que recebe um vetor com datas e determina o dia da semana bem como faz uma contagem das semanas.
- A função deve retornar um data.frame com as seguintes colunas: date, weekday, week. Exemplo:

date	weekday	week
2020-01-01	4	1
2020-01-02	5	1
2020-01-03	6	1
2020-01-04	7	1
2020-01-05	1	2
2020-01-06	2	2
2020-01-07	3	2
2020-01-08	4	2
2020-01-09	5	2
2020-01-10	6	2
2020-01-11	7	2
2020-01-12	1	3

4. Camada de lógica

Crie um script chamado “main.r”, o script deve fazer as seguintes tarefas:

- Carregar a função criada anteriormente.
- Carregar o script DataLoader.r.
- Utilizar a função criada para determinar qual a semana associada a cada registro na base “covid data”. (dica utilizar join).
- Excluir a informação da última semana pois essa pode estar incompleta.

5. Camada de usuário

Crie uma rotina que cria 2 gráficos e salva eles no diretório “Graficos”.

- Gráfico 1: utilizando os dados agrupados, crie um gráfico de linha do total de mortes por covid- 19, com uma série para cada país selecionado. (eixo x: Semana, eixo y: Total de mortes).
- Gráfico 2: utilizando os dados agrupados, crie um gráfico de linha das novas de mortes por covid-19, com uma série para cada país selecionado. (eixo x: Semana, eixo y: Novas mortes)
- Salve os dois gráficos no diretório “Graficos”. (Configurações recomendadas: scale=1, units = “in”, dpi = 300, width = 10.4, height = 5.85).