

# Introdução à programação com R

## Controle de fluxo e repetições



# Tópicos desta aula

- Loops de repetição: for
- Controle de fluxo: if

# Loops de repetição: for

# Loops de repetição: for

- O `for` permite repetir uma mesma ação várias vezes, dentro de um determinado escopo.
- O escopo é o contexto em que a ação será repetida
- A ação deverá ser aplicada utilizando o elemento do escopo
- Observe a estrutura estrutura:

```
for(ESCOPO) {  
    # Ação que será feita  
}
```

O escopo também tem uma estrutura importante:

```
for(posicao_do_elemento in conjunto_de_valores) {  
    # Ação que será feita  
}
```

Exemplo 1: Queremos iterar entre os números 1 até 10, e "printando" o número no console.

```
for(numero in 1:10){  
  print(numero)  
}
```

```
## [1] 1  
## [1] 2  
## [1] 3  
## [1] 4  
## [1] 5  
## [1] 6  
## [1] 7  
## [1] 8  
## [1] 9  
## [1] 10
```

Exemplo 2: Queremos calcular a média de atraso de saída para cada dia dos voos de janeiro.

```
# Carregando os dados -----  
library(readr)  
library(dplyr)  
base_de_dados <- read_csv2("../dados/voos_de_janeiro.csv")  
  
#---- repeticao!  
for(dia_filtrar in 1:31) {  
  base_filtrada <- filter(base_de_dados, dia == dia_filtrar)  
  media_atraso_saida <- round(  
    mean(base_filtrada$atraso_saida, na.rm = TRUE)  
    , 1)  
  
  print(  
    paste0(  
      "A média de atraso de saída no dia ",  
      dia_filtrar,  
      " é de ",  
      media_atraso_saida,  
  
      " minutos."  
    )  
  )  
}  
}
```

```
## [1] "A média de atraso de saída no dia 1 é de 11.5 minutos."  
## [1] "A média de atraso de saída no dia 2 é de 13.9 minutos."  
## [1] "A média de atraso de saída no dia 3 é de 11 minutos."  
## [1] "A média de atraso de saída no dia 4 é de 9 minutos."
```

# Controle de fluxo: if

# Controle de fluxo: if

- O `if` permite executar um código condicional.
- Em português, pense no `if` como a palavra "SE", e o `else` como "SE NÃO":

```
if (condicao) { # Se a condição for verdadeira  
  # o código que é executado quando a condição é verdadeira (TRUE)  
} else { # Se não  
  # o código que é executado quando a condição é falsa (FALSE)  
}
```



- Podemos usar o `else if` para colocar outras condições:

```
if (condicao) {  
  # faz coisa tal  
} else if (outra_condicao) {  
  # faz outra coisa  
} else {  
  # o que fazer com o que não foi verdadeiro em todas as condições anteriores  
}
```

Imagine que você trabalhe em uma loja e a meta de vendas é de 500 reais, e você quer saber se falta muito para você atingir a meta! O objeto `valor_vendido` representa o quanto você já vendeu, e o controle de fluxo (com `if`) pode nos ajudar a descobrir se a meta foi atingida.

```
valor_vendido <- 300

if (valor_vendido >= 500) {
  print("Parabéns, você atingiu a meta!")
} else if (valor_vendido >= 400 & valor_vendido < 500) {
  print("Olha só, você está perto de atingir a meta! Falta pouco!")
} else {
  print("Hum.. ainda falta bastante para a meta ser atingida.")
}
```

```
## [1] "Hum.. ainda falta bastante para a meta ser atingida."
```