

Control Flow

Marcella Pedro

25/11/2021

Controle de Fluxo (If... Else..)

Nesse exercício vamos aprender como fazemos para tomar decisões através do código, direcionar quando um bloco de código deve ser executado, ou até mesmo determinar o valor da coluna de um dataset através de condições.

If else simples

Utilizado em situações onde você precisa executar um bloco de código específico dependendo da situação.

Precisamos apresentar uma condição ao if, que se resulte em verdadeiro (TRUE) vai executar o primeiro bloco, caso contrário o segundo bloco será executado.

```
graduacao <- "Engenharia"

if (graduacao == "Química") {
  print("Graduado em Química")
} else {
  print("Não graduado em Química.")
}
```

```
## [1] "Não graduado em Química."
```

If else no dataframe

Importando um dataset de exemplo

```
library(readr)
library(dplyr)

df <- read.csv("~/Documentos/exemplo.csv")

df %>% head()
```

```
##      nome idade  genero      profissao
## 1 marcella   26  feminino analista de dados
## 2 rafael    28  masculino analista de sistemas
## 3 ana vitoria 24  feminino analista contabil
## 4 maria     16  feminino  jovem aprendiz
## 5 joao      17  masculino estagiario
```

Vamos criar uma flag que indica se a pessoa é menor de idade. A função `if_else` simula o mesmo comportamento do `if else` acima, mas ao invés de executar um trecho de código diferente, atribui um valor diferente à nova coluna.

```
df <- df %>% mutate(
  flg_menor = if_else(idade<18,"Menor de idade", "Maior de idade")
)
df %>% head()
```

```
##      nome idade  genero      profissao    flg_menor
## 1 marcella   26  feminino analista de dados Maior de idade
## 2 rafael    28  masculino analista de sistemas Maior de idade
## 3 ana vitoria 24  feminino analista contabil Maior de idade
## 4 maria     16  feminino  jovem aprendiz Menor de idade
## 5 joao      17  masculino   estagiario Menor de idade
```

If else com mais de uma condição

Por vezes não temos apenas uma condição quando verdadeira executa um bloco e caso contrário outro bloco, pode ser que seja uma situação complexa que exija mais de duas soluções.

```
idade = 30

if(idade < 18){
  print("Menor de idade")
}else if(idade >60){
  print("Aposentado")
}else{
  print("Adulto")
}
```

```
## [1] "Adulto"
```

If else com mais de uma condição simultânea

Às vezes para que uma decisão seja tomada, duas condições precisam ser avaliadas em conjunto.

```
temperatura = 30

if(temperatura >=18 && temperatura <=25){
  print("Clima ideal")
}else{
  print("Poderia ser melhor :(")
}
```

```
## [1] "Poderia ser melhor :("
```

Agora utilizando a função `if_else` tem uma pegadinha, se usarmos os operadores `&&` (and) e `||` (or) temos um resultado inesperado, traz apenas um resultado quando esperamos um resultado para cada nota. É preciso usar apenas `&` e `|` para termos o resultado esperado.

```
nota <- c(4,6,7,3,10)

resultado_errado <- if_else(nota >= 5 && nota <= 10, "Nota Azul", "Nota Vermelha")
print(resultado_errado)
```

```
## [1] "Nota Vermelha"
```

```
resultado_correto <- if_else(nota >= 5 & nota <= 10, "Nota Azul", "Nota Vermelha")
print(resultado_correto)
```

```
## [1] "Nota Vermelha" "Nota Azul"      "Nota Azul"      "Nota Vermelha"
## [5] "Nota Azul"
```

Case_when

Quando queremos mais de uma condição na onde normalmente usaríamos a função `if_else`, podemos então usar o `case_when`.

A condição final utiliza `TRUE` e se comporta como o `else`.

```
df <- df %>% mutate(
  status = case_when(idade<18 ~ "Menor de idade",
                     idade>60 ~ "Aposentado",
                     TRUE ~ "Adulto")
)
df %>% head()
```

```
##      nome idade  genero      profissao    flg_menor
## 1 marcella   26  feminino analista de dados Maior de idade
## 2 rafael     28  masculino analista de sistemas Maior de idade
## 3 ana vitoria 24  feminino analista contabil Maior de idade
## 4 maria      16  feminino   jovem aprendiz Menor de idade
## 5 joao       17  masculino   estagiario Menor de idade
##      status
## 1      Adulto
## 2      Adulto
## 3      Adulto
## 4 Menor de idade
## 5 Menor de idade
```

Dica Bônus: %in%

Quando temos muitas comparações, pode ser cansativo e repetitivo criar o código.

```
df %>% filter(profissao=="analista de dados" |
              profissao=="analista de sistemas" |
              profissao=="analista contabil") %>%
  head
```

```
##      nome idade  genero      profissao      flg_menor status
## 1  marcella   26  feminino analista de dados Maior de idade Adulto
## 2   rafael   28  masculino analista de sistemas Maior de idade Adulto
## 3  ana vitoria 24  feminino  analista contabil Maior de idade Adulto
```

Uma forma mais simplificada de realizar a mesma comparação é utilizando o %in%

```
df %>% filter(profissao %in% c("analista de dados",
                              "analista de sistemas",
                              "analista contabil")) %>%
  head
```

```
##      nome idade  genero      profissao      flg_menor status
## 1  marcella   26  feminino analista de dados Maior de idade Adulto
## 2   rafael   28  masculino analista de sistemas Maior de idade Adulto
## 3  ana vitoria 24  feminino  analista contabil Maior de idade Adulto
```

Ou caso a ideia seja pegar tudo que seja diferente dessas opções, em outras palavras não seja essas opções (not in), basta adicionar o ponto de exclamação para inverter.

```
df %>% filter(!profissao %in% c("analista de dados",
                              "analista de sistemas",
                              "analista contabil")) %>%
  head
```

```
##      nome idade  genero      profissao      flg_menor      status
## 1  maria    16  feminino jovem aprendiz Menor de idade Menor de idade
## 2   joao    17  masculino  estagiario Menor de idade Menor de idade
```