

Aula 08-05-2025

Prof^a. Izadora S. de C.

Estruturas de Controle em R: Continuação

Tipo	Usado para...
if, else	tomar decisões
for	repetir uma ação para vários valores
while	repetir enquanto uma condição for válida

#Exemplo com if

```
x <- 25

if (x > 20) {
  print("Está quente")
}
```

```
## [1] "Está quente"
```

#FAÇA VOCÊ MESMO

```
x <- 18
if(x > 20) {
  print("Está quente")
}
```

#Exemplo com if e else

```
temperatura <- 23

if (temperatura > 28) {
  print("Está quente!")
} else {
  print("Temperatura agradável.")
}
```

```
## [1] "Temperatura agradável."
```

#FAÇA VOCÊ MESMO

Crie um vetor com 7 temperaturas e use uma estrutura `ifelse` para classificar como “frio” (< 18), “agradável” (18–28), ou “quente” (> 28).

```
#Exemplo com 'for'

#Criando vetor
dias <- c("Seg", "Ter", "Qua")

#Aplicando for para imprimir na tela os dias da semana
for (a in dias) {
  print(a)
}
```

```
## [1] "Seg"
## [1] "Ter"
## [1] "Qua"
```

```
#FAÇA VOCÊ MESMO
```

```
#Crie vetor com todos os dias da semana
```

```
#Exemplo com while
```

```
x <- 1

while (x <= 3) {
  print(x)
  x <- x + 1
}
```

```
## [1] 1
## [1] 2
## [1] 3
```

Funções (criadas)

As funções são criadas com `function()`.

Podemos passar quantos argumentos quiser.

Pode usar `return()` ou deixar o último valor ser retornado automaticamente.

`sapply()` é útil para aplicar uma função a vetores/listas.

```
#Função soma
```

```
#Aplicando função
```

```
# Função para calcular a amplitude térmica do dia
amplitude_termica <- function(temp_max, temp_min) {
  return(temp_max - temp_min)
}

# Usando a função
amplitude_termica(32.5, 21.8)
```

#REPRODUZA

Função que classifica temperaturas como "Frio", "Agradável" ou "Quente"

```
classificar_temp <- function(temp) {  
  if (temp < 20) {  
    return("Frio")  
  } else if (temp <= 27) {  
    return("Agradável")  
  } else {  
    return("Quente")  
  }  
}  
  
# Aplicando a função a um vetor  
temperaturas <- c(25.2, 26.1, 27.0, 26.8)  
  
sapply(temperaturas, classificar_temp)
```

#REPRODUZA

Desafio:

#Criar Função para converter temperatura
#formula: $C = K - 273.15$

Manipulação de Dados com R

Etapas e processos relacionados a manipulação de dados no R

- Carregar dados;
- Explorar a estrutura e tipo dos dados;
- Limpeza e tratamento dos dados;
- Análise Exploratória.

Pacotes necessário:

1. `tidyverse` (`dplyr` e `ggplot2`) : para manipulação e visualização de dados de forma eficiente.
2. `lubridate` : para facilitar o trabalho com datas.

```
# Instale os pacotes se necessário  
# install.packages(c("tidyverse", "lubridate"))  
  
library(tidyverse)  
library(lubridate)
```

```
# Exemplo 01
```

```
dados_meteorologicos <- tibble(  
  data = seq.Date(from = as.Date("2024-01-01"), to = as.Date("2024-12-31"), by = "day"),  
  temperatura_max = rnorm(366, mean = 30, sd = 5),  
  temperatura_min = rnorm(366, mean = 20, sd = 5),  
  precipitacao = rpois(366, lambda = 5)  
)
```

```
# Reproduza o dataframe do Exemplo 01
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --  
## v dplyr      1.1.4      v readr      2.1.5  
## v forcats    1.0.0      v stringr    1.5.1  
## v ggplot2    3.5.1      v tibble     3.2.1  
## v lubridate  1.9.4      v tidyr      1.3.1  
## v purrr      1.0.4
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()     masks stats::lag()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
```

```
dados_meteorologicos <- tibble(  
  data = seq.Date(from = as.Date("2024-01-01"),  
                    to = as.Date("2024-12-31"), by = "day"),  
  temperatura_max = rnorm(366, mean = 30, sd = 5),  
  temperatura_min = rnorm(366, mean = 20, sd = 5),  
  precipitacao = rpois(366, lambda = 5)  
)
```

```
# Visualizar os primeiros registros da tabela criada
```

```
# Verificar estrutura
```

```
#str
```

```
#glimpse
```

```
# Resumo estatístico
```

```
#summary()
```

```
# install.packages(skimr)
```

```
# library(skimr)
```

Principais funções para manipulação de dados:

- `select()`: para selecionar variáveis;
- `filter()`: para filtrar observações;

- `arrange()`: para classificar variáveis;
- `mutate()`: para criar e transformar variáveis;
- `group_by()`: para agrupar observações;
- `summarise()`: para resumir os dados com medidas estatísticas descritivas.

Fluxo de trabalho com *pipe*

Aprimorando fluxo de trabalho com *pipe* `%>%` ou `|>` (atalho: Ctrl + Shift + M). Com o *pipe* é possível o encadeamento de várias funções, eliminando a necessidade de criar objetos para armazenar resultados intermediários. Sendo assim, uma ferramenta poderosa para melhorar o fluxo de trabalho na análise e manipulação de dados.

```
#Sem pipe
sqrt(sum(1:50))

#Com pipe
1:50 |>
  sum() |>
  sqrt()
```

Aplicando principais funções de manipulação:

```
# Selecionar apenas as colunas de data e temperatura máxima
#select()

# dados_temMax <- select(dados_meteorologicos, data, temperatura_max)

dados_meteorologicos %>%
  select(data, temperatura_max)
```

```
## # A tibble: 366 x 2
##   data      temperatura_max
##   <date>          <dbl>
## 1 2024-01-01          29.9
## 2 2024-01-02          27.3
## 3 2024-01-03          30.8
## 4 2024-01-04          40.8
## 5 2024-01-05          26.3
## 6 2024-01-06          38.3
## 7 2024-01-07          32.9
## 8 2024-01-08          31.9
## 9 2024-01-09          39.3
## 10 2024-01-10         21.0
## # i 356 more rows
```

```
# Filtrar dias com precipitação acima de 10 mm
#filter()

dados_meteorologicos %>%
  filter(precipitacao > 10)
```

```
## # A tibble: 3 x 4
##   data      temperatura_max temperatura_min precipitacao
##   <date>      <dbl>          <dbl>          <int>
## 1 2024-08-02      31.5            21.3            14
## 2 2024-09-20      27.8            18.6            11
## 3 2024-10-18      31.7            24.7            13
```

```
# Filtrar dados de janeiro
dados_meteorologicos %>%
  filter(month(data) == 1)
```

```
## # A tibble: 31 x 4
##   data      temperatura_max temperatura_min precipitacao
##   <date>      <dbl>          <dbl>          <int>
## 1 2024-01-01      29.9            24.0             5
## 2 2024-01-02      27.3            29.2             5
## 3 2024-01-03      30.8            27.8             3
## 4 2024-01-04      40.8            17.4             8
## 5 2024-01-05      26.3            19.7             6
## 6 2024-01-06      38.3            25.5             3
## 7 2024-01-07      32.9            26.3             7
## 8 2024-01-08      31.9            19.9             5
## 9 2024-01-09      39.3            14.1             2
## 10 2024-01-10     21.0            19.7             7
## # i 21 more rows
```

```
# Ordenar os dados pela temperatura máxima, da maior para a menor
#arrange()
dados_meteorologicos %>%
  arrange(desc(temperatura_max))
```

```
## # A tibble: 366 x 4
##   data      temperatura_max temperatura_min precipitacao
##   <date>      <dbl>          <dbl>          <int>
## 1 2024-03-04      44.2            16.3             5
## 2 2024-06-26      42.3            15.6             9
## 3 2024-11-11      42.2            14.6             7
## 4 2024-03-29      41.7            26.1             3
## 5 2024-09-06      41.6            16.4             4
## 6 2024-03-17      41.1            18.9             4
## 7 2024-01-04      40.8            17.4             8
## 8 2024-07-18      40.7            14.1             2
## 9 2024-08-27      40.4            20.6             3
## 10 2024-02-15     40.2            22.3             6
## # i 356 more rows
```

```
# Ordenar pelas datas mais recentes
dados_meteorologicos %>%
  arrange(desc(data))
```

```
## # A tibble: 366 x 4
##   data      temperatura_max temperatura_min precipitacao
```

```
##      <date>                <dbl>                <dbl>                <int>
##  1 2024-12-31                32.2                15.0                 6
##  2 2024-12-30                25.6                19.7                 1
##  3 2024-12-29                24.4                18.8                 4
##  4 2024-12-28                27.3                24.9                 8
##  5 2024-12-27                28.2                25.3                 3
##  6 2024-12-26                32.4                26.5                 1
##  7 2024-12-25                30.9                23.0                 6
##  8 2024-12-24                26.8                22.0                 4
##  9 2024-12-23                28.5                16.3                 6
## 10 2024-12-22                24.1                26.0                 8
## # i 356 more rows
```

```
# Criar uma nova coluna com a média diária de temperatura
#mutate()
dados_meteorologicos %>%
  mutate(temp_media = (temperatura_max + temperatura_min) / 2)
```

```
## # A tibble: 366 x 5
##   data      temperatura_max temperatura_min precipitacao temp_media
##   <date>                <dbl>                <dbl>                <int>      <dbl>
##  1 2024-01-01                29.9                24.0                 5        26.9
##  2 2024-01-02                27.3                29.2                 5        28.3
##  3 2024-01-03                30.8                27.8                 3        29.3
##  4 2024-01-04                40.8                17.4                 8        29.1
##  5 2024-01-05                26.3                19.7                 6        23.0
##  6 2024-01-06                38.3                25.5                 3        31.9
##  7 2024-01-07                32.9                26.3                 7        29.6
##  8 2024-01-08                31.9                19.9                 5        25.9
##  9 2024-01-09                39.3                14.1                 2        26.7
## 10 2024-01-10                21.0                19.7                 7        20.3
## # i 356 more rows
```

```
# Adicionar o mês como nova coluna
dados_meteorologicos %>%
  mutate(mes = month(data, label = TRUE))
```

```
## # A tibble: 366 x 5
##   data      temperatura_max temperatura_min precipitacao mes
##   <date>                <dbl>                <dbl>                <int> <ord>
##  1 2024-01-01                29.9                24.0                 5 jan
##  2 2024-01-02                27.3                29.2                 5 jan
##  3 2024-01-03                30.8                27.8                 3 jan
##  4 2024-01-04                40.8                17.4                 8 jan
##  5 2024-01-05                26.3                19.7                 6 jan
##  6 2024-01-06                38.3                25.5                 3 jan
##  7 2024-01-07                32.9                26.3                 7 jan
##  8 2024-01-08                31.9                19.9                 5 jan
##  9 2024-01-09                39.3                14.1                 2 jan
## 10 2024-01-10                21.0                19.7                 7 jan
## # i 356 more rows
```

```
# Adicionar o ano como nova coluna
```

```
# Precipitação total por mês  
#group_by() e #summarise()
```

```
dados_meteorologicos %>%  
  mutate(mes = month(data, label = TRUE)) %>%  
  group_by(mes) %>%  
  summarise(precipitacao_total = sum(precipitacao))
```

```
## # A tibble: 12 x 2  
##   mes      precipitacao_total  
##   <ord>                <int>  
## 1 jan                160  
## 2 fev                137  
## 3 mar                136  
## 4 abr                151  
## 5 mai                168  
## 6 jun                162  
## 7 jul                145  
## 8 ago                168  
## 9 set                163  
## 10 out               146  
## 11 nov               140  
## 12 dez               140
```

```
# Temperatura máxima média por trimestre
```

```
dados_meteorologicos %>%  
  mutate(trimestre = quarter(data)) %>%  
  group_by(trimestre) %>%  
  summarise(temp_max_media = mean(temperatura_max))
```

```
## # A tibble: 4 x 2  
##   trimestre temp_max_media  
##   <int>          <dbl>  
## 1     1           30.2  
## 2     2           31.1  
## 3     3           29.2  
## 4     4           30.6
```

```
# Calcular temperatura média diária e total de chuva por mês, ordenando por maior chuva
```

```
dados_meteorologicos %>%  
  mutate(temp_media = (temperatura_max + temperatura_min) / 2,  
         mes = month(data, label = TRUE)) %>%  
  group_by(mes) %>%  
  summarise(  
    media_temp = mean(temp_media),  
    chuva_total = sum(precipitacao)  
  ) %>%  
  arrange(desc(chuva_total))
```

```
## # A tibble: 12 x 3
```



```
##   mes   media_temp chuva_total
##   <ord>      <dbl>      <int>
## 1 mai        25.3        168
## 2 ago        24.0        168
## 3 set        23.8        163
## 4 jun        25.8        162
## 5 jan        24.6        160
## 6 abr        24.8        151
## 7 out        25.2        146
## 8 jul        24.6        145
## 9 nov        25.5        140
## 10 dez       25.0        140
## 11 fev       24.2        137
## 12 mar       25.6        136
```

Trabalhando com datas com o pacote lubridate :

No R, datas são tratadas como um tipo especial de objeto, como classe date. Há várias formas de criar objetos dessa classe com o pacote lubridate.

```
# Usando a função as_date():
data <- lubridate::as_date('2024-12-20')
class(data)
```

```
## [1] "Date"
```

A função `as_date()` assume que a ordem segue o padrão da língua inglesa:

- ano-mês-dia (ymd).

No pacote lubridate, existem funções para todas as ordens possíveis:

- `ymd()`, `dmy()`, `mdy()`, `myd()`, `ydm()` etc.

```
ymd('2024-12-20')
```

```
## [1] "2024-12-20"
```

```
dmy('20-12-2024')
```

```
## [1] "2024-12-20"
```

```
dmy('20122024')
```

```
## [1] "2024-12-20"
```

Se for trabalhar com data e horário, basta usar as funções do tipo:

- `ymd_h()`, `ymd_hm()`, `ymd_hms()`.

```
ymd_hms(20241220093010)
```

```
## [1] "2024-12-20 09:30:10 UTC"
```

O pacote ‘lubridate’ fornece diversas funções para extrair os componentes de um objeto da classe `date`.

1. `year()` - extrai o ano.
2. `month()` - extrai o mês.
3. `day()` - extrai o dia.
4. `hour()` - extrai a hora.
5. `minute()` - extrai os minutos.
6. `second()` - extrai os segundos.

```
# Exemplos:
```

```
year(data)
```

```
## [1] 2024
```

```
month(data)
```

```
## [1] 12
```

```
day(data)
```

```
## [1] 20
```

Para criar uma sequência de datas, podemos usar a função `seq.Date()` da base do R em combinação com funções do pacote ‘lubridate’:

```
# Primeiro, vamos definir a data inicial e final.
```

```
data_inicial <- ymd('2024-01-01')
```

```
data_final <- today() # A função today() do pacote 'lubridate'  
# retorna a data atual no formato Date.
```

```
# Criando uma sequência de datas diária.
```

```
seq_diaria <- seq(data_inicial, data_final, by = 'day')  
# print(seq_diaria)
```

```
# Criar a sequência de datas mensal.
```

```
seq_mensal <- seq(data_inicial, data_final, by = 'month')  
print(seq_mensal)
```

```
## [1] "2024-01-01" "2024-02-01" "2024-03-01" "2024-04-01" "2024-05-01"  
## [6] "2024-06-01" "2024-07-01" "2024-08-01" "2024-09-01" "2024-10-01"  
## [11] "2024-11-01" "2024-12-01" "2025-01-01" "2025-02-01" "2025-03-01"  
## [16] "2025-04-01" "2025-05-01"
```

Análise Exploratória.

Visualizações

```
# Dados agrupados por mês
precip_mensal <- dados_meteorologicos %>%
  mutate(mes = month(data, label = TRUE)) %>%
  group_by(mes) %>%
  summarise(precipitacao_total = sum(precipitacao))
```

```
#FAÇA VOCÊ MESMO
```

```
# Gráfico de barras
```

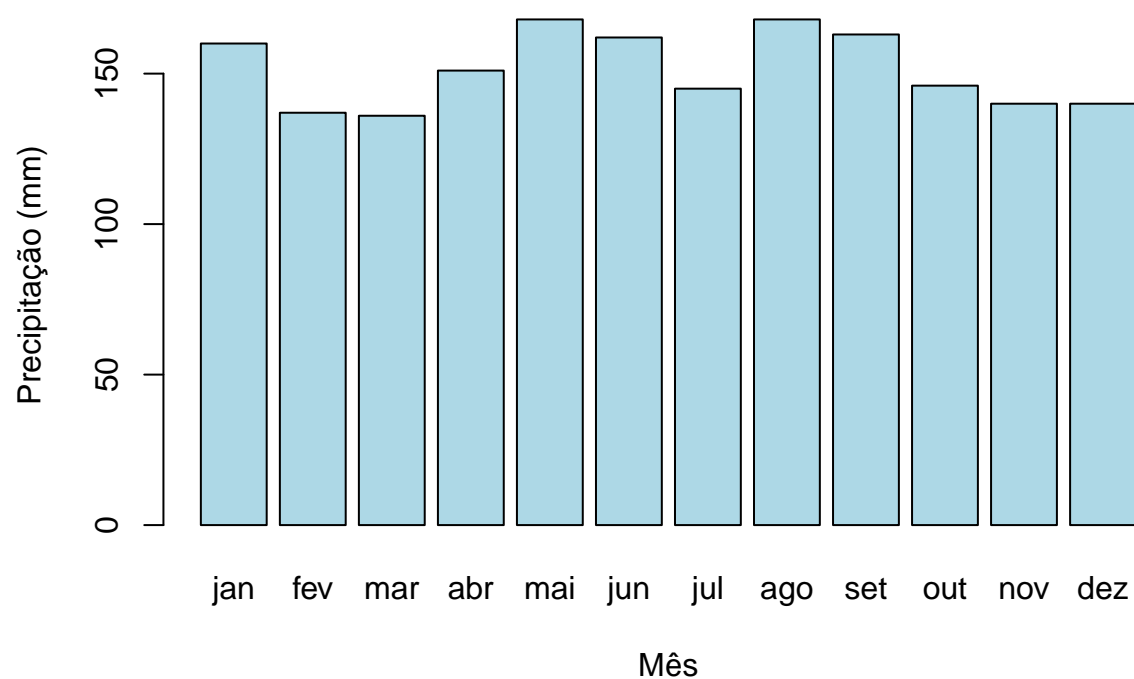
```
#r base
str(precip_mensal)
```

```
## tibble [12 x 2] (S3: tbl_df/tbl/data.frame)
## $ mes : Ord.factor w/ 12 levels "jan"<"fev"<"mar"<...: 1 2 3 4 5 6 7 8 9 10 ...
## $ precipitacao_total: int [1:12] 160 137 136 151 168 162 145 168 163 146 ...
```

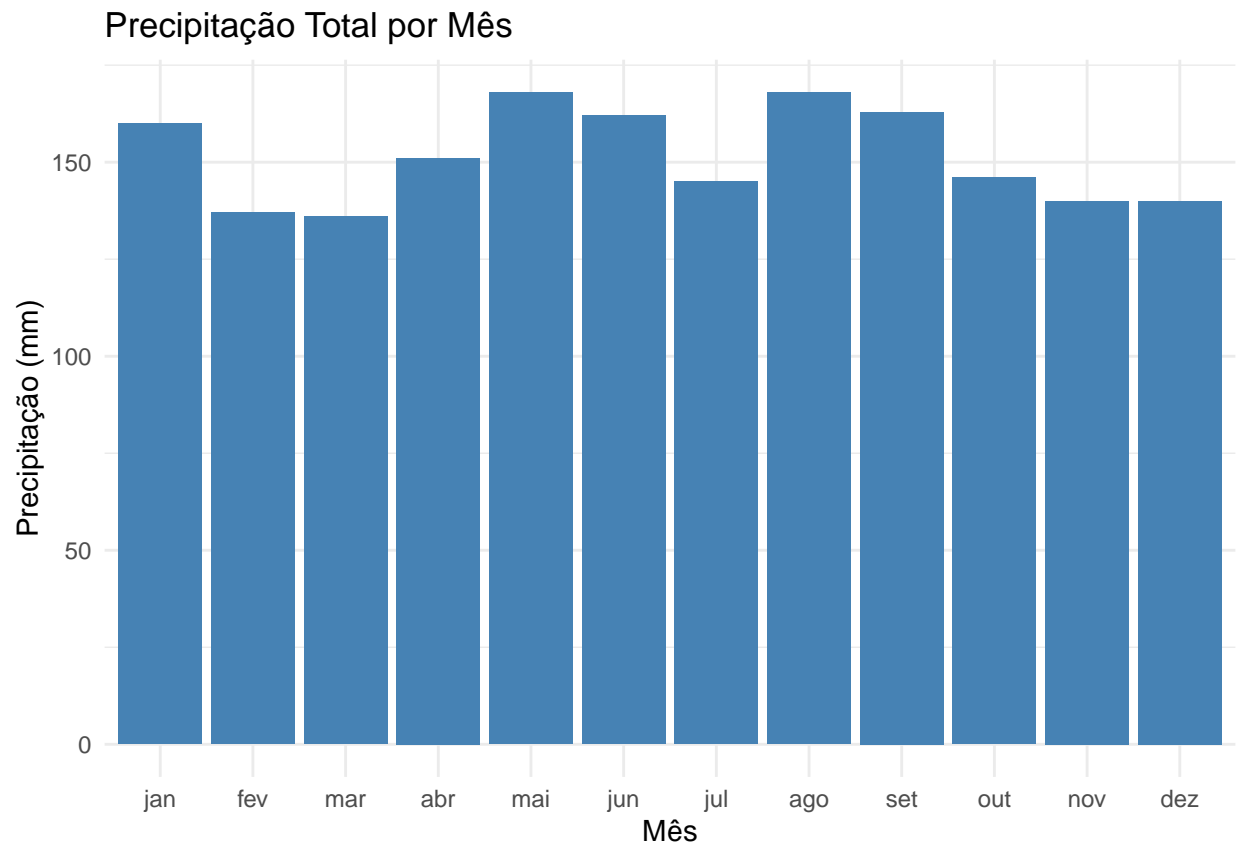
```
meses <- as.character(precip_mensal$mes)
valores <- precip_mensal$precipitacao_total

barplot(height = precip_mensal$precipitacao_total,
  names.arg = precip_mensal$mes,
  col = "lightblue",
  main = "Precipitação Total por Mês (Base R)",
  xlab = "Mês",
  ylab = "Precipitação (mm)")
```

Precipitação Total por Mês (Base R)



```
#com ggplot2
ggplot(precip_mensal, aes(x = mes, y = precipitacao_total)) +
  geom_col(fill = "steelblue") +
  labs(title = "Precipitação Total por Mês",
       x = "Mês", y = "Precipitação (mm)") +
  theme_minimal()
```



Referências:

- Pacote tinytex para exportar relatório em formato pdf
- Rmarkdown
- Quarto