# Atividade Avaliativa

Introdução à Linguagem R

Ana Maria Alves da Silva

2024-07-13

### Importante

Entregue qualquer coisa que você consiga fazer. É normal que encontre dificuldades. Qualquer coisa que seja entregue será passível de análise e receber nota.

### Análise de Dado do Portal COVID

Nesta atividade estão sendo analisados os dados casos e óbitos de COVID em Goiânia no ano de 2021. Os dados foram obtidos do site do Ministério da Saúde.

### Item 1. Importe os dados de COVID-19 do ano de 2021, obtidos no site do Ministério da Saúde, ou baixada da página do curso.

**Solução:**

Para importar os dados podemos utilizar, após a instalação, o pacote *readxl* e a função *read_excel* deste pacote desde que os dados a serem importados estejam em formato excel. Para utilizar esta função basta passarmos o caminho de onde está o arquivo excel que desejamos importar juntamente com o nome do arquivo. Além disso, como vamos utilizar esse arquivo no decorrer da atividade, iremos salva-lo em uma váriavel chamada df. Veja comandos abaixo.

```
df <- read_excel("/Users/anamaria/especializacao/modulo_3/atividade/df.xlsx")
```

Observe que para melhor leitura do texto, renomeei o arquivo excel fornecido para df.

## Considerando a base toda (Todo o território nacional)

### Item 2. Realize a limpeza dos dados de todas as colunas (*NA's, Datas, etc*). Use as funções e as ferramentas trabalhadas em aula.

**Solução:** Note que a limpeza dos dados é uma parte essencial para que análise de dados seja efetiva pois dados faltantes (Na's) e dados em formatos incorretos podem interferir ou impedir que os cálculos sejam realizados ou levar a informações inveridicas e imprecisas. Logo, antes que qualquer tipo de análise seja realizada é recomendado que seja realizado a limpeza dos dados para que as análise posteriores sejam assertivas e coerentes.

- Verificando se df é um dataframe:

```
is_data_frame <- is.data.frame(df)
print(is_data_frame)
```

```
## [1] TRUE
```

Observe que se df não fosse um dataframe poderiamos utilizar o comando a baixo para transformar df em um dataframe. Como, no nosso caso, df já é um dataframe o comando não está sendo executado.

```
df <- as.data.frame(df)
```

- Renomear as colunas do dataframe para remover espaços e acentuação:

```
names(df)
```

```
## [1] "Info"
## [2] "UF"
## [3] "Município"
## [4] "Metro/Interior"
## [5] "Ano_Semana"
## [6] "Casos Acumulados"
## [7] "Casos novos notificados na semana epidemiológica"
## [8] "Óbitos Acumulados"
## [9] "Óbitos novos notificados na semana epidemiológica"
```

```
names(df)[names(df) == "Município"] <- "Municipio"
names(df)[names(df) == "Casos novos notificados na semana epidemiológica"] <- "Casos_novos_notificados_s
names(df)[names(df) == "Óbitos novos notificados na semana epidemiológica"] <- "Obitos_novos_notificados
names(df)[names(df) == "Casos Acumulados"] <- "Casos_Acumulados"
names(df)[names(df) == "Óbitos Acumulados"] <- "Obitos_Acumulados"
```

- Verificar o tipo de dado de cada coluna:

```
str(df)
```

```
## tibble [296,323 x 9] (S3: tbl_df/tbl/data.frame)
##  $ Info                                          : chr [1:296323] "COVID19Casos" "COVID19Casos" "COVI
##  $ UF                                            : chr [1:296323] "AC" "AC" "AC" "AC" ...
##  $ Municipio                                     : chr [1:296323] "Acrelândia" "Assis Brasil" "Brasil
##  $ Metro/Interior                                : chr [1:296323] "Interior" "Interior" "Interior" "I
##  $ Ano_Semana                                    : chr [1:296323] "53/2021" "53/2021" "53/2021" "53/1
##  $ Casos_Acumulados                              : num [1:296323] 563 762 1305 491 306 ...
##  $ Casos_novos_notificados_semana_epidemiologica : num [1:296323] 0 0 7 0 1 3 1 30 0 0 ...
##  $ Obitos_Acumulados                             : num [1:296323] 12 9 22 8 8 72 16 26 1 15 ...
##  $ Obitos_novos_notificados_semana_epidemiologica: num [1:296323] 0 0 0 0 0 0 0 0 0 0 ...
```

Note que a colunas do tipo Ano_Semana faz referência a um período temporal mas seu tipo de dado está no formato string ou caracter. Como em demais questões da atividade será necessário realizar algumas operações envolvendo ANO irei criar uma nova coluna chamada ANO na qual eu irei extrair o ano correspondente da coluna Ano_Semana usando a função **substr**. Faremos o mesmo para determinar a semana epidemológica.

```r
df$Ano <- ifelse(nchar(df$Ano_Semana) == 7,
                 substr(df$Ano_Semana, 4, 7),
                 ifelse(nchar(df$Ano_Semana) == 6, substr(df$Ano_Semana, 3, 6), NA))

df$Semana <- ifelse(nchar(df$Ano_Semana) == 7,
                 substr(df$Ano_Semana, 1, 2),
                 ifelse(nchar(df$Ano_Semana) == 6, substr(df$Ano_Semana, 1, 1), NA))
```

- Vamos agora, verificar se há valores ausentes. Inicialmente vamos verificar a quantidade de valores ausentes no dataframe através de:

```r
sum(is.na(df))
```

```
## [1] 1113
```

Como obtido pelo comando acima, sabemos que há mais de mil valores ausentes. Vamos agora, verificar os valores ausentes por colunas do dataframe.

```r
colSums(is.na(df))
```

```
##                                        Info
##                                           0
##                                          UF
##                                           0
##                                   Municipio
##                                        1113
##                                Metro/Interior
##                                           0
##                                   Ano_Semana
##                                           0
##                                Casos_Acumulados
##                                           0
##  Casos_novos_notificados_semana_epidemiologica
##                                           0
##                                Obitos_Acumulados
##                                           0
## Obitos_novos_notificados_semana_epidemiologica
##                                           0
##                                         Ano
##                                           0
##                                      Semana
##                                           0
```

Logo, a única coluna do dataframe que possui valores ausentes é a coluna **Município**. Agora, vamos identificar se há alguma caracteristica comum aos municípios que possuem valores ausentes para que possamos identificar a melhor maneira de lidar com eles, isso é, remover do dataframe ou substitui-los por alguma informação relevante. No caso de optarmos por substituição dos NA's, é importante verificamos qual o tipo de dado da coluna município para que a substituição seja realizada de forma coerente.

```r
missing_values = subset(df, is.na(Municipio))
head(missing_values, 20)
```

```
## # A tibble: 20 x 11
##    Info         UF    Municipio `Metro/Interior` Ano_Semana Casos_Acumulados
##    <chr>        <chr> <chr>     <chr>            <chr>                  <dbl>
##  1 COVID19Casos AL    <NA>      Interior         53/2021                    8
##  2 COVID19Casos BA    <NA>      Interior         53/2021                 7011
##  3 COVID19Casos CE    <NA>      Interior         53/2021                 5973
##  4 COVID19Casos ES    <NA>      Interior         53/2021                 1915
##  5 COVID19Casos GO    <NA>      Interior         53/2021                    0
##  6 COVID19Casos MA    <NA>      Interior         53/2021                    0
##  7 COVID19Casos MG    <NA>      Interior         53/2021                 3126
##  8 COVID19Casos MT    <NA>      Interior         53/2021                    0
##  9 COVID19Casos PB    <NA>      Interior         53/2021                    0
## 10 COVID19Casos PE    <NA>      Interior         53/2021                    0
## 11 COVID19Casos PI    <NA>      Interior         53/2021                    0
## 12 COVID19Casos PR    <NA>      Interior         53/2021                 3181
## 13 COVID19Casos RJ    <NA>      Interior         53/2021                    0
## 14 COVID19Casos RN    <NA>      Interior         53/2021                    0
## 15 COVID19Casos RO    <NA>      Interior         53/2021                    0
## 16 COVID19Casos RR    <NA>      Interior         53/2021                 1818
## 17 COVID19Casos RS    <NA>      Interior         53/2021                    0
## 18 COVID19Casos SC    <NA>      Interior         53/2021                10260
## 19 COVID19Casos SE    <NA>      Interior         53/2021                    0
## 20 COVID19Casos SP    <NA>      Interior         53/2021                  213
## # i 5 more variables: Casos_novos_notificados_semana_epidemiologica <dbl>,
## #   Obitos_Acumulados <dbl>,
## #   Obitos_novos_notificados_semana_epidemiologica <dbl>, Ano <chr>,
## #   Semana <chr>
```

Note que em várias linhas correspondentes aos valores ausentes na coluna Município, as colunas casos acumulados, Casos novos notificados na semana epidemiológica, Óbitos Acumulados e Óbitos novos notificados na semana epidemiológica possuem valor 0, nesses casos, optarei por excluir essas linhas do dataframe. Já para a situação onde essas colunas não são nulas há algumas possibilidades a serem consideradas, uma delas seria substituir pelo valor que mais aparece (moda) ou substituir pelo valor que está no meio (mediana), no entanto como ainda não vimos esses métodos estatísticos ainda eu não irei aplica-los. Eu optarei por criar uma nova categoria chamda "Não informado" para realizar a substituição desses dados ausentes.

```
df$Municipio[is.na(df$Municipio) & df$Casos_Acumulados != 0] <- "Não informado"
colSums(is.na(df))
```

```
##                                        Info
##                                           0
##                                          UF
##                                           0
##                                   Municipio
##                                         706
##                                Metro/Interior
##                                           0
##                                   Ano_Semana
##                                           0
##                              Casos_Acumulados
##                                           0
##  Casos_novos_notificados_semana_epidemiologica
##                                           0
```

```
##                               Obitos_Acumulados
##                                               0
## Obitos_novos_notificados_semana_epidemiologica
##                                               0
##                                             Ano
##                                               0
##                                          Semana
##                                               0
```

```
df <- na.omit(df)
colSums(is.na(df))
```

```
##                                             Info
##                                                0
##                                               UF
##                                                0
##                                        Municipio
##                                                0
##                                   Metro/Interior
##                                                0
##                                       Ano_Semana
##                                                0
##                                  Casos_Acumulados
##                                                0
##  Casos_novos_notificados_semana_epidemiologica
##                                                0
##                               Obitos_Acumulados
##                                                0
## Obitos_novos_notificados_semana_epidemiologica
##                                                0
##                                              Ano
##                                                0
##                                           Semana
##                                                0
```

- Verficar se há linhas duplicadas no dataframe:

```
duplicados <- duplicated(df)
ha_duplicados <- any(duplicados)
print(ha_duplicados)
```

```
## [1] FALSE
```

Note que não há colunas duplicadas, caso houvesse na a váriavel ha_duplicados retornaria "TRUE". Se houvesse dados duplicados poderiamos remove-los do dataframe usando:

```
df <- df[!duplicated(df), ]
```

Dessa forma, realizamos a limpeza dos dados.

### Item 3. Use a função "factor" para categorizar a coluna "Metro/Interior" no data frame.

**Solução:**

Note que a coluna "Metro/Interior" é uma categorização da região, sendo metro para região metropolitana da UF ou Interior para uma região no interior da UF, logo podemos usar a função factor para categorizar essa coluna.

```
df$`Metro/Interior` <- factor(df$`Metro/Interior`, levels = c("Reg. Metropolitana", "Interior"),
          labels = c("Reg. Metropolitana", "Interior"))
is_factor <- is.factor(df$`Metro/Interior`)
print(is_factor)
```

```
## [1] TRUE
```

### Item 4. Realize um resumo dos dados utilizando a função summary.

**Solução:**

```
summary_df <- summary(df)
print(summary_df)
```

```
##      Info                UF             Municipio
##  Length:295617      Length:295617      Length:295617
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##             Metro/Interior    Ano_Semana        Casos_Acumulados
##  Reg. Metropolitana: 20458   Length:295617      Min.   :     1
##  Interior          :275159   Class :character   1st Qu.:   319
##                              Mode  :character   Median :   734
##                                                 Mean   :  3054
##                                                 3rd Qu.:  1898
##                                                 Max.   :977918
##  Casos_novos_notificados_semana_epidemiologica Obitos_Acumulados
##  Min.   :   -7.00                              Min.   :    0.00
##  1st Qu.:    1.00                              1st Qu.:    6.00
##  Median :    4.00                              Median :   15.00
##  Mean   :   21.62                              Mean   :   83.42
##  3rd Qu.:   13.00                              3rd Qu.:   40.00
##  Max.   :59430.00                              Max.   :39561.00
##  Obitos_novos_notificados_semana_epidemiologica     Ano
##  Min.   :   0.000                               Length:295617
##  1st Qu.:   0.000                               Class :character
##  Median :   0.000                               Mode  :character
##  Mean   :   0.795
##  3rd Qu.:   1.000
##  Max.   :4109.000
##     Semana
```

```
##  Length:295617
##  Class :character
##  Mode  :character
##
##
##
```

Note que a função summary está nos trazendo que existem valores negativos na coluna Casos_novos_notificados_semana_epic
de fator, veja:

```
casos_negativos <- df[df$Casos_novos_notificados_semana_epidemiologica < 0,]
print(casos_negativos$Casos_novos_notificados_semana_epidemiologica)
```

```
## [1] -1 -2 -7 -2 -1 -1 -2
```

Como não tem como ter tido casos negativos de Covid podemos considerar que foi um erro de digitação ou
um erro do sistema e vamos substituir esses valores por 0.

```
df$Casos_novos_notificados_semana_epidemiologica[df$Casos_novos_notificados_semana_epidemiologica < 0] <
```

Validando novamente a summarização:

```
summary_df_new <- summary(df)
print(summary_df_new)
```

```
##      Info                UF             Municipio
##  Length:295617      Length:295617      Length:295617
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##             Metro/Interior    Ano_Semana        Casos_Acumulados
##  Reg. Metropolitana: 20458   Length:295617      Min.   :     1
##  Interior          :275159   Class :character   1st Qu.:   319
##                              Mode  :character   Median :   734
##                                                 Mean   :  3054
##                                                 3rd Qu.:  1898
##                                                 Max.   :977918
##  Casos_novos_notificados_semana_epidemiologica Obitos_Acumulados
##  Min.   :    0.00                              Min.   :    0.00
##  1st Qu.:    1.00                              1st Qu.:    6.00
##  Median :    4.00                              Median :   15.00
##  Mean   :   21.62                              Mean   :   83.42
##  3rd Qu.:   13.00                              3rd Qu.:   40.00
##  Max.   :59430.00                              Max.   :39561.00
##  Obitos_novos_notificados_semana_epidemiologica     Ano
##  Min.   :   0.000                                  Length:295617
##  1st Qu.:   0.000                                  Class :character
##  Median :   0.000                                  Mode  :character
##  Mean   :   0.795
##  3rd Qu.:   1.000
```

```
##  Max.   :4109.000
##     Semana
##  Length:295617
##  Class :character
##  Mode  :character
##
##
##
```

### Item 5. Filtre os dados para a cidade de Goiânia e mostre as 10 primeiras linhas do data frame.

**Solução:**

```
df_goiania <- df[df$Municipio == "Goiânia", ]
print(head(df_goiania, 10))
```

```
## # A tibble: 10 x 11
##    Info         UF    Municipio `Metro/Interior`   Ano_Semana Casos_Acumulados
##    <chr>        <chr> <chr>     <fct>              <chr>                  <dbl>
##  1 COVID19Casos GO    Goiânia   Reg. Metropolitana 53/2021                79301
##  2 COVID19Casos GO    Goiânia   Reg. Metropolitana 1/2021                 81980
##  3 COVID19Casos GO    Goiânia   Reg. Metropolitana 2/2021                 86475
##  4 COVID19Casos GO    Goiânia   Reg. Metropolitana 3/2021                 88919
##  5 COVID19Casos GO    Goiânia   Reg. Metropolitana 4/2021                 92011
##  6 COVID19Casos GO    Goiânia   Reg. Metropolitana 5/2021                 95799
##  7 COVID19Casos GO    Goiânia   Reg. Metropolitana 6/2021                 98670
##  8 COVID19Casos GO    Goiânia   Reg. Metropolitana 7/2021                100279
##  9 COVID19Casos GO    Goiânia   Reg. Metropolitana 8/2021                103917
## 10 COVID19Casos GO    Goiânia   Reg. Metropolitana 9/2021                106951
## # i 5 more variables: Casos_novos_notificados_semana_epidemiologica <dbl>,
## #   Obitos_Acumulados <dbl>,
## #   Obitos_novos_notificados_semana_epidemiologica <dbl>, Ano <chr>,
## #   Semana <chr>
```

## Considerando os dados Filtrados de Goiânia

### Item 6. Calcule e exiba o número de casos e de óbitos acumulados de COVID-19 no ano de 2021.

**Solução:**

Note que no dataframe *df_goiania*, onde está filtrado apenas os dados de covid referente a Goiânia há apenas casos de covid referente ao Ano de 2021. Logo, para obter o números de casos de de óbitos acumlados no ano de 2021 basta usarmos o colSums() para somar os dados das respectivas colunas. Note ainda, que no exercício 2 desta atividade nós verificamos que essas colunas são to tipo numerica, logo não é necessário realizarmos nenhum tipo de alteração.

```
## [1] "O total de casos acumulados de Covid19 em Goiânia no Ano de 2021 foi de  8419629 casos."
```

```
## [1] "Enquanto o total de óbitos acumulados foi 267354 no Ano de 2021."
```

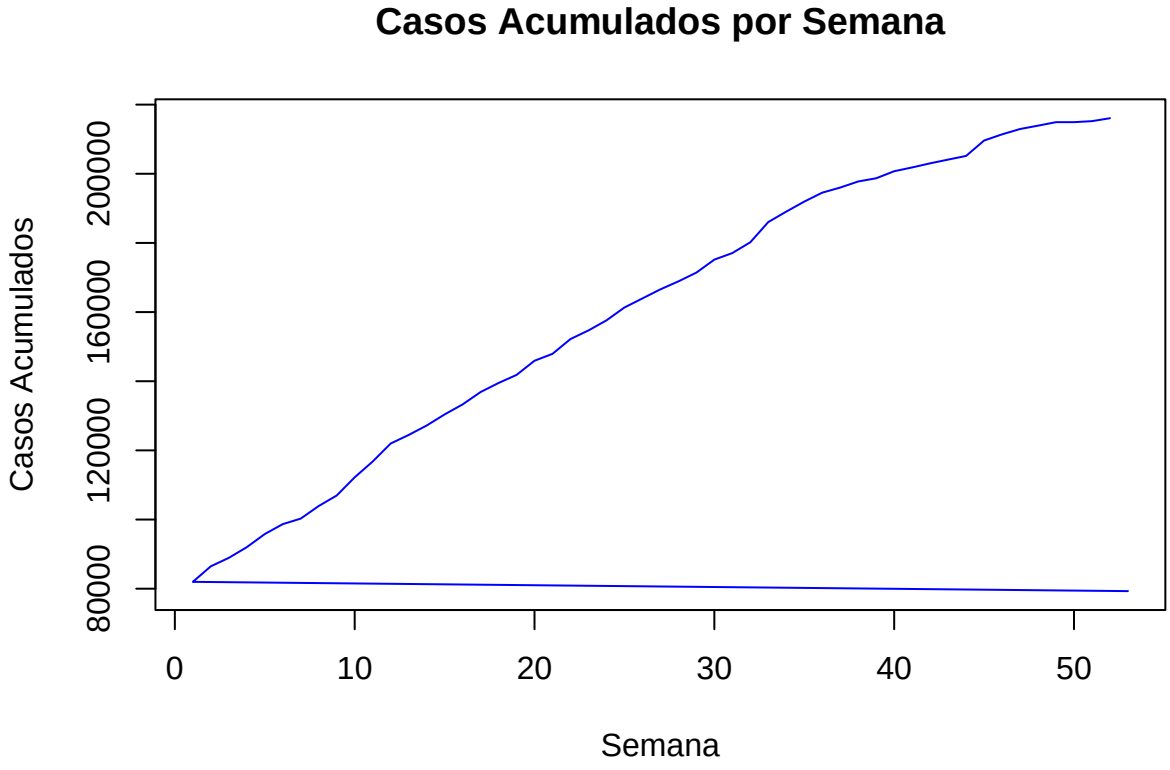
**Item 7. Gere e exiba um gráfico de linha de casos acumulados de COVID-19 por semana em 2021.**

**Solução:**

Para gerar e exibir o gráfico de linhas solicitados, iremos filtrar a Semana, através da coluna que criamos no exercício 2, e a coluna de casos acumulados. Depois iremos usar a função plot.

```
x_1 <- df_goiania$Semana
y_1 <- df_goiania$Casos_Acumulados

plot(x_1, y_1, type = "l",
     main = "Casos Acumulados por Semana",
     xlab = "Semana",
     ylab = "Casos Acumulados",
     col = "blue",
     lwd = 1)
```



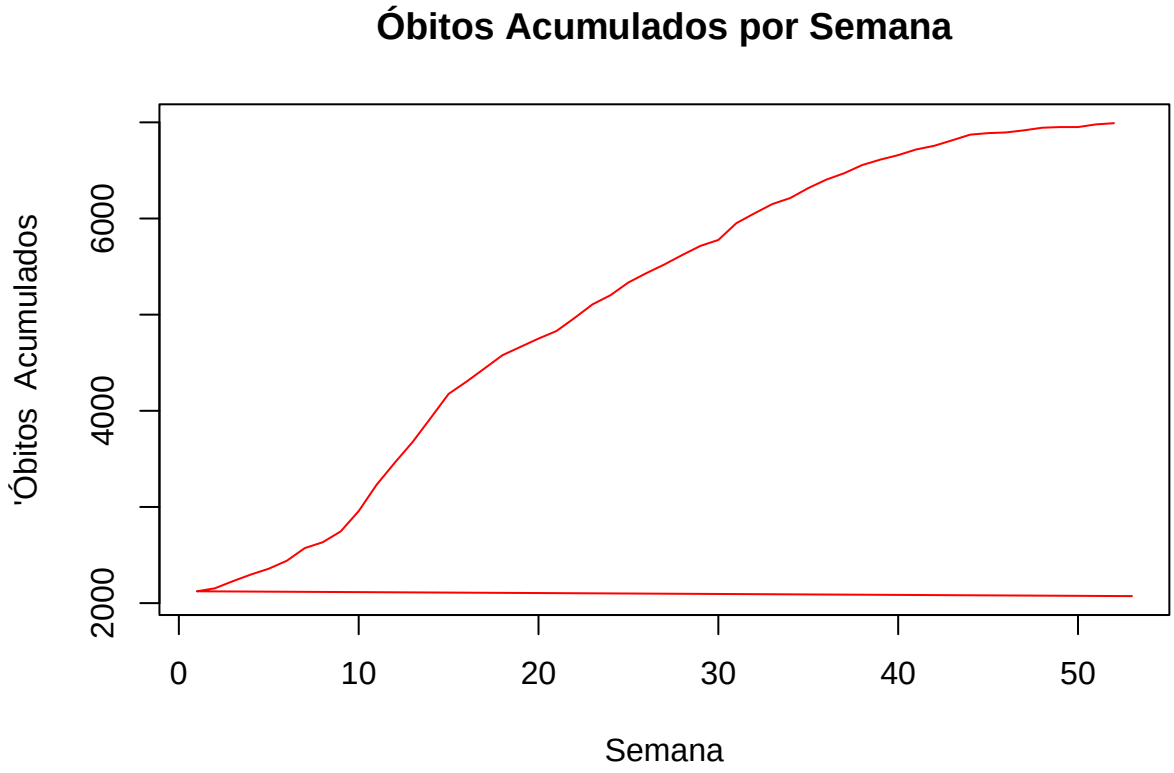

**Item 8. Gere e exiba um gráfico de linha de óbitos acumulados de COVID-19 por semana em 2021.**

**Solução:**

Para gerar e exibir o gráfico de linhas solicitados, iremos filtrar a Semana, através da coluna que criamos no exercício 2, e a coluna de obtos acumulados. Depois iremos usar a função plot.

```r
x_2 <- df_goiania$Semana
y_2 <- df_goiania$Obitos_Acumulados

plot(x_2, y_2, type = "l",
     main = "Óbitos Acumulados por Semana",
     xlab = "Semana",
     ylab = "'Óbitos  Acumulados",
     col = "red",
     lwd = 1)
```




## Bônus

**Item 9. Crie uma nova coluna denominada "Regiao", partir da coluna "UF". Agrupe as UF segundo suas regiões específicas: Norte, Nordeste, Centro-Oeste, Sul e Sudeste.**

**Solução:**

Irei realizar a criação dessa coluna usando um ifelse para identificar se a UF correspondente a Região está na coluna UF e categorizar a partir dessa filtragem.

```r
df$Regiao <- ifelse(df$UF %in% c("AC", "AP", "AM", "PA", "RO", "RR", "TO"), "Norte",
                    ifelse(df$UF %in% c("AL", "BA", "CE", "MA", "PB", "PI", "PE", "RN", "SE"), "Nordeste
```

```
                            ifelse(df$UF %in% c("DF", "GO", "MT", "MS"), "Centro-Oeste",
                                   ifelse(df$UF %in% c("PR", "RS", "SC"), "Sul",
                                          ifelse(df$UF %in% c("ES", "MG", "RJ", "SP"), "Sudeste", NA))))

df_filtrado <- df[, c("UF", "Regiao")]
head(df_filtrado, 10)
```

```
## # A tibble: 10 x 2
##    UF    Regiao
##    <chr> <chr>
##  1 AC    Norte
##  2 AC    Norte
##  3 AC    Norte
##  4 AC    Norte
##  5 AC    Norte
##  6 AC    Norte
##  7 AC    Norte
##  8 AC    Norte
##  9 AC    Norte
## 10 AC    Norte
```