Lista 1

2025-11-03

Pacotes necessários

Para essa lista de exercícios serão necessários os pacotes *data.table* e *dplyr*, para leitura e manipulação das bases de dados, respoectivamente.

```
# instale os pacotes com os comando abaixo
# install.packages("tidyverse") # obs: o pacote dplyr faz parte do pacote tidyverse
# install.packages("data.table")

# apos a instalação, importe os pacotes
library(tidyverse)
library(data.table)
```

Exercicio 1 : Leitura dos dados

A função `fread` da biblioteca `data.table` é uma versão otimizada das funções `read.csv` e `read.csv2` para grandes bases de dados.

a) Leia novamente as bases de dados da CGIL de 2024 e 2025 utilizando a função `fread`. (Guarde cada uma das bases em uma variavel)

Dica : Consulte o `help(fread)` para mais informações sobre a leitura.

```
dados_2024 <- fread("CGIL_CNIg_2024.csv", encoding="UTF-8")
dados_2025 <- fread("CGIL_CNIg_jan-set2025.csv", encoding="UTF-8")
```

b) Junte as duas bases de dados em uma só variável.

```
dados <- rbind(dados_2024, dados_2025)
```

Exercicio 2 : Limpeza e manipulação dos dados

O pacote `dyplr` é focado em manipulação de bases de dados. A pesar de ter um grande numero de funções, a maoria delas tem nomes intuitivos, por exemplo:

- `filter` : filtra as linhas de acordo com as condições dadas;
- `select` : seleciona as colunas indicadas;
- `group_by` : agrupa os dados de acordo com as colunas escolhidas;
- `count` : conta a quantidade de elementos na coluna;
- `summarise` : agrega os dados das colunas indicadas de acordo com as medidas escolhidas.

Esses comandos podem ser encadeados para recrear as tabelas vistas ao longo do curso, como por exemplo os dados filtrados

```r
dados24 <- read.csv2("CGIL_CNIg_2024.csv", fileEncoding = "UTF-8")
dados25 <- read.csv2("CGIL_CNIg_jan-set2025.csv", fileEncoding = "UTF-8")
dados <- rbind(dados24, dados25)

dados_final <- dados %>%
  filter(modalidade == "CGIL", andamento == "DEFERIDO") %>%
  filter((ano == 2024 & mes == 9) | (ano == 2025 & mes %in% c(8,9)))
```

Ou, para replicar a tabela de frequência por gênero

```r
dados_final %>%
  select(ano, mes, genero) %>%
  group_by(ano, mes, genero) %>%
  summarise(n = n(),.groups = "drop") %>%
  pivot_wider(names_from = c(mes, ano),
              names_prefix = "d_",
              values_from = n)
```

```
## # A tibble: 2 x 4
##   genero d_9_2024 d_8_2025 d_9_2025
##   <chr>     <int>    <int>    <int>
## 1 F           291      312      588
## 2 M          2298     3703     4084
```

```r
paises <- dados_final %>%
  select(ano, mes, pais) %>%
  group_by(ano, mes, pais) %>%
  summarise(n = n(),.groups = "drop") %>%
  pivot_wider(names_from = c(mes, ano),
              names_prefix = "d_",
              values_from = n)
```

a) Utilize as funções do pacote `dyplr` para recriar a tabela de frquência de paises por mes e ano.

```r
paises <- paises %>%
  mutate(total = d_9_2024 + d_8_2025 + d_9_2025)
```

b) A partir da tabela encontrada na letra a) crie uma nova coluna com o total de frequencias ao longo dos anos.

```r
paises %>%
  arrange(desc(total)) %>%
  slice_head(n = 10)
```

c) Agora selecione os 10 paises de maior frequência total. (Dica: `help(arange)`)

```
## # A tibble: 10 x 5
##    pais           d_9_2024 d_8_2025 d_9_2025 total
##    <chr>             <int>    <int>    <int> <int>
##  1 CHINA               695     1291     1158  3144
##  2 FILIPINAS           193      261      396   850
##  3 ESTADOS UNIDOS      181      227      212   620
##  4 ÍNDIA               151      124      309   584
```

```
##  5 ITÁLIA              111      126      309   546
##  6 GRÃ-BRETANHA         83      138      102   323
##  7 BANGLADESH            6      249       65   320
##  8 ALEMANHA             80      117      113   310
##  9 MÉXICO               58      126      109   293
## 10 FRANÇA               61      102       99   262
```