```
## ----eval=FALSE-------------------------------------------------------------------------------
## # Usando read.csv()
## dados_read_csv <- read.csv("dados.csv")
##
## # Usando read_csv() do pacote readr
## library(readr)
## dados_readr <- read_csv("dados.csv")


## ----eval=FALSE-------------------------------------------------------------------------------
## library(readxl)
## dados <- read_excel("arquivo.xlsx")


## ----eval=FALSE-------------------------------------------------------------------------------
## setwd("C:/MeuDiretorio")


## ----eval=FALSE-------------------------------------------------------------------------------
## resultado <- f(g(x))
##
## x %>%
##   g() %>%
##   f()


## ----echo=FALSE, message=FALSE, warning=FALSE-------------------------------------------------
library(tidyverse)


## ---------------------------------------------------------------------------------------------
table1
table2
```

```
table4a


## ------------------------------------------------------------------------------------------------
---------------
table2 %>%
  pivot_wider(names_from="type", values_from="count")


## ------------------------------------------------------------------------------------------------
---------------
table4a %>%
  pivot_longer(cols = c(`1999`, `2000`), names_to = "year", values_to = "cases")


library(gapminder)
head(gapminder)


glimpse(gapminder)


# Select ----------------------------------------------------------------

## # Selecionando as colunas pelo nome
gapminder %>%
  select(year, country, lifeExp, gdpPercap)

# Selecionando apenas as colunas numéricas
gapminder %>%
  select(where(is.numeric))

# Selecionando colunas que começam com "co"
gapminder %>%
  select(starts_with("co"))


gapminder_character <- gapminder %>%
  select(where(is.character))
```

```
# arrange ------------------------------------------------------------


gapminder %>%
  select(year, country, lifeExp, gdpPercap, pop) %>%
  arrange(country)


gapminder %>%
  select(year, country, lifeExp, gdpPercap, pop) %>%
  arrange(year, desc(lifeExp))


gapminder %>%
  select(-continent)


gapminder %>%
  select(year, country, lifeExp, gdpPercap, pop) %>%
  arrange(year, desc(lifeExp)) %>%
  filter(country == "Brazil" | country == "Argentina")


# mutate -------------------------------------------------------------


gapminder_total_gdp <- gapminder %>%
  select(country, year, lifeExp, gdpPercap, pop) %>%
  mutate(total_gdp = gdpPercap * pop)


# summarize ----------------------------------------------------------


gapminder %>%
  summarise(mean_lifeExp = mean(lifeExp, na.rm = TRUE))
```

```r
gapminder %>%
  group_by(continent) %>%
  summarise(mean_lifeExp = mean(lifeExp, na.rm = TRUE))


# group by ------------------------------------------------------------

gapminder %>%
  select(country, continent, year, lifeExp, gdpPercap) %>%
  filter(year == 2007) %>% # apenas os dados para o ano de 2007
  mutate(gdp = gdpPercap / 1000) %>% # representa o PIB per capita em milhares
  group_by(continent) %>% # agrupar os dados por continente
  summarise(mean_lifeExp = mean(lifeExp, na.rm = TRUE), # média da expectativa de vida
            mean_gdp = mean(gdp, na.rm = TRUE)) %>% #média do PIB per capita em bilhões
  arrange(desc(mean_lifeExp))




# Exercicios ----------------------------------------------------------

library(tidyverse)
billboard


## resposta a
billboard %>%
  pivot_longer(
    cols = starts_with("wk"),
    names_to = "week",
    values_to = "rank"
  )


## resposta b
## billboard %>%
##   pivot_longer(
##     cols = starts_with("wk"),
```

```
##     names_to = "week",
##     values_to = "rank",
##     values_drop_na = TRUE
##   )


## resposta c
## billboard_longer <- billboard %>%
##   pivot_longer(
##     cols = starts_with("wk"),
##     names_to = "week",
##     values_to = "rank",
##     values_drop_na = TRUE
##   ) %>%
##   mutate(
##     week = parse_number(week)
##   )


## resposta d
## billboard_longer %>%
##   group_by(track) %>%
##   summarise(n=n()) %>%
##   arrange(desc(n))


## resposta e
## billboard_longer %>%
##   group_by(track, date.entered) %>%
##   summarise(n=n()) %>%
##   filter(n==10) %>%
##   arrange(date.entered)
```