Comparação R x Python

Gabriella de Oliveira Argenton 255677

```
library(tidyverse)
```

```
Warning: pacote 'ggplot2' foi compilado no R versão 4.5.2
```

```
Warning: pacote 'tidyr' foi compilado no R versão 4.5.2
```

```
Warning: pacote 'purrr' foi compilado no R versão 4.5.2
```

```
Warning: pacote 'dplyr' foi compilado no R versão 4.5.2
```

```
Warning: pacote 'stringr' foi compilado no R versão 4.5.2
```

```
-- 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   1.6.0
v ggplot2   4.0.0     v tibble    3.3.0
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.2.0
-- 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 beco
```

```
library(dplyr)
library(microbenchmark)
```

```
Warning: pacote 'microbenchmark' foi compilado no R versão 4.5.2
```

```r
set.seed(123)

n <- 1e6
df_big <- tibble(
  grupo = sample(c("A", "B", "C", "D"), n, replace = TRUE),
  x     = sample(1:100,  n, replace = TRUE),
  y     = sample(1:1000, n, replace = TRUE)
)

# Operação a ser medida
resumo_por_grupo <- function(df) {
  df |>
    group_by(grupo) |>
    summarise(y_media = mean(y), .groups = "drop")
}

# Teste simples
resumo_por_grupo(df_big)
```

```
# A tibble: 4 x 2
  grupo y_media
  <chr>   <dbl>
1 A        501.
2 B        502.
3 C        501.
4 D        500.
```

```r
# Benchmark
mb <- microbenchmark(
  resumo_por_grupo(df_big),
  times = 10
)

mb
```

```
Unit: milliseconds
                    expr    min      lq     mean  median      uq     max neval
 resumo_por_grupo(df_big) 9.5845 10.9482 11.83474 11.9362 12.6181 13.7454    10
```

```r
summary(mb)
```

```
                      expr    min      lq     mean  median      uq     max
1 resumo_por_grupo(df_big) 9.5845 10.9482 11.83474 11.9362 12.6181 13.7454
  neval
1    10
```

```python
import numpy as np
import pandas as pd
import timeit

np.random.seed(123)

n = 1_000_000
df_big = pd.DataFrame({
    "grupo": np.random.choice(["A","B","C","D"], size=n),
    "x": np.random.randint(1, 101, size=n),
    "y": np.random.randint(1, 1001, size=n)
})

def resumo_por_grupo(df):
    return (
        df
        .groupby("grupo", as_index=False)
        ["y"]
        .mean()
        .rename(columns={"y": "y_media"})
    )

# Teste simples
print(resumo_por_grupo(df_big))
```

```
  grupo     y_media
0     A  501.151220
1     B  499.561209
2     C  499.571024
3     D  499.884592
```

```python
# Benchmark: executa 10 vezes e mede o tempo total
tempo = timeit.timeit(
```

```
    stmt="resumo_por_grupo(df_big)",
    globals=globals(),
    number=10
)

print("Tempo total para 10 execuções (Python):", tempo, "segundos")
```

```
Tempo total para 10 execuções (Python): 0.24649519997183233 segundos
```

```
print("Tempo médio por execução:", tempo/10, "segundos")
```

```
Tempo médio por execução: 0.024649519997183233 segundos
```