

desafio01

2025-09-24

```
# -----  
# Benchmark de leitura do flights.csv (US DOT Flight Delays)  
# -----  
  
# Caminho do ZIP baixado do Kaggle (ajuste se necessário)  
zip_path <- "flights.csv.zip"  
  
if (!file.exists(zip_path)) {  
  stop("Arquivo 'flights.csv.zip' não encontrado. Baixe do Kaggle e ajuste 'zip_path'.")  
}  
  
# 1) Descompactar para um arquivo temporário (fread não lê .zip direto)  
tmp_dir <- tempdir()  
unzipped_files <- unzip(zip_path, exdir = tmp_dir)  
# Se houver mais de um arquivo no zip, tentamos pegar o .csv principal  
csv_path <- unzipped_files[grepl("\\.csv$", unzipped_files, ignore.case = TRUE)][1]  
if (is.na(csv_path)) stop("Nenhum .csv encontrado dentro do zip.")  
  
message("CSV extraído para: ", csv_path)
```

CSV extraído para: C:/Users/pepem/AppData/Local/Temp/RtmpuuuAAn/flights.csv

```
# 2) Pacotes usados  
suppressPackageStartupMessages({  
  if (!requireNamespace("readr", quietly = TRUE)) {  
    stop("Pacote 'readr' não instalado. Instale com install.packages('readr').")  
  }  
  if (!requireNamespace("data.table", quietly = TRUE)) {  
    stop("Pacote 'data.table' não instalado. Instale com install.packages('data.table').")  
  }  
})  
  
# 3) Função auxiliar para cronometrar de forma justa  
time_it <- function(expr) {  
  gc() # limpa um pouco a memória entre tentativas  
  invisible(NULL)  
  t <- system.time({  
    obj <- eval.parent(substitute(expr))  
    # opcionalmente verifica dimensões para garantir que leu algo  
    print(paste("Linhas x Colunas:", paste(dim(obj), collapse = " x ")))  
    rm(obj); invisible(gc())  
  })  
  unname(t["elapsed"])  
}
```

```

# 4) Rodar múltiplas repetições e pegar a mediana (mais robusto)
n <- 3 # aumente para 5 ou 7 se quiser
run_many <- function(fun_call, label) {
  times <- numeric(n)
  for (i in seq_len(n)) {
    message(sprintf("[%s] repetição %d/%d ...", label, i, n))
    times[i] <- time_it(fun_call)
  }
  list(label = label, times = times, median = median(times))
}

```

```

# 5) Executar os três métodos
res_base <- run_many(read.csv(csv_path),
  label = "base::read.csv")

```

```
## [base::read.csv] repetição 1/3 ...
```

```
## [1] "Linhas x Colunas: 5819079 x 31"
```

```
## [base::read.csv] repetição 2/3 ...
```

```
## [1] "Linhas x Colunas: 5819079 x 31"
```

```
## [base::read.csv] repetição 3/3 ...
```

```
## [1] "Linhas x Colunas: 5819079 x 31"
```

```

res_readr <- run_many(readr::read_csv(csv_path, show_col_types = FALSE, progress = FALSE),
  label = "readr::read_csv")

```

```
## [readr::read_csv] repetição 1/3 ...
```

```
## [1] "Linhas x Colunas: 5819079 x 31"
```

```
## [readr::read_csv] repetição 2/3 ...
```

```
## [1] "Linhas x Colunas: 5819079 x 31"
```

```
## [readr::read_csv] repetição 3/3 ...
```

```
## [1] "Linhas x Colunas: 5819079 x 31"
```

```

res_datatable <- run_many(data.table::fread(csv_path, showProgress = FALSE),
  label = "data.table::fread")

```

```
## [data.table::fread] repetição 1/3 ...
```

```
## [1] "Linhas x Colunas: 5819079 x 31"
```

```
## [data.table::fread] repetição 2/3 ...

## [1] "Linhas x Colunas: 5819079 x 31"

## [data.table::fread] repetição 3/3 ...

## [1] "Linhas x Colunas: 5819079 x 31"

# 6) Mostrar resultados
cat("\n=== Tempos (segundos) por repetição ===\n")

##
## === Tempos (segundos) por repetição ===

print(sprintf("%s: %s", res_base$label, paste(round(res_base$times, 3), collapse = ", ")))

## [1] "base::read.csv: 26.8, 0.06, 0.04"

print(sprintf("%s: %s", res_readr$label, paste(round(res_readr$times, 3), collapse = ", ")))

## [1] "readr::read_csv: 6.59, 0.06, 0.06"

print(sprintf("%s: %s", res_datatable$label, paste(round(res_datatable$times, 3), collapse = ", ")))

## [1] "data.table::fread: 0.77, 0.05, 0.05"

cat("\n=== Medianas (segundos) ===\n")

##
## === Medianas (segundos) ===

med <- c(`base::read.csv`      = res_base$median,
        `readr::read_csv`    = res_readr$median,
        `data.table::fread`  = res_datatable$median)
print(round(med, 3))

##      base::read.csv      readr::read_csv data.table::fread
##           0.06           0.06           0.05

cat("\nMétodo mais rápido no seu computador:", names(med)[which.min(med)], "\n")

##
## Método mais rápido no seu computador: data.table::fread
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

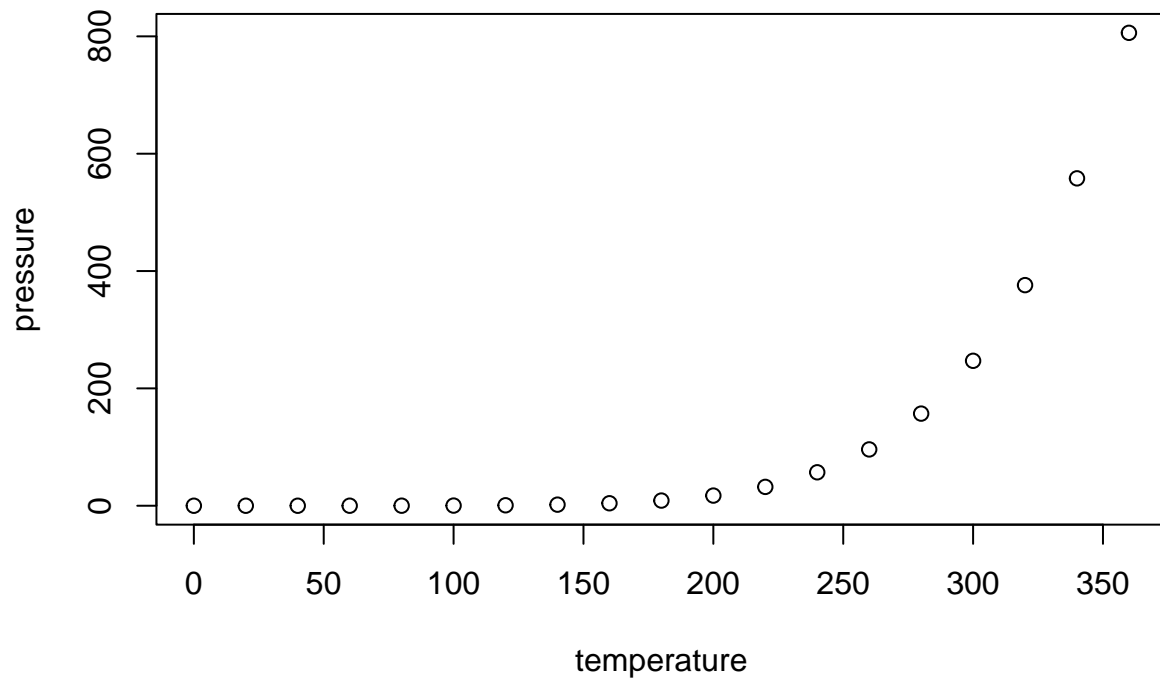
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.    : 2.00
## 1st Qu.:12.0    1st Qu.: 26.00
##  Median :15.0    Median : 36.00
##   Mean  :15.4    Mean     : 42.98
## 3rd Qu.:19.0    3rd Qu.: 56.00
##   Max.  :25.0    Max.     :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.