Algorithme d'optimisation de stockage

Yoann Bonnet, Victorien Leconte, Hugo Picard

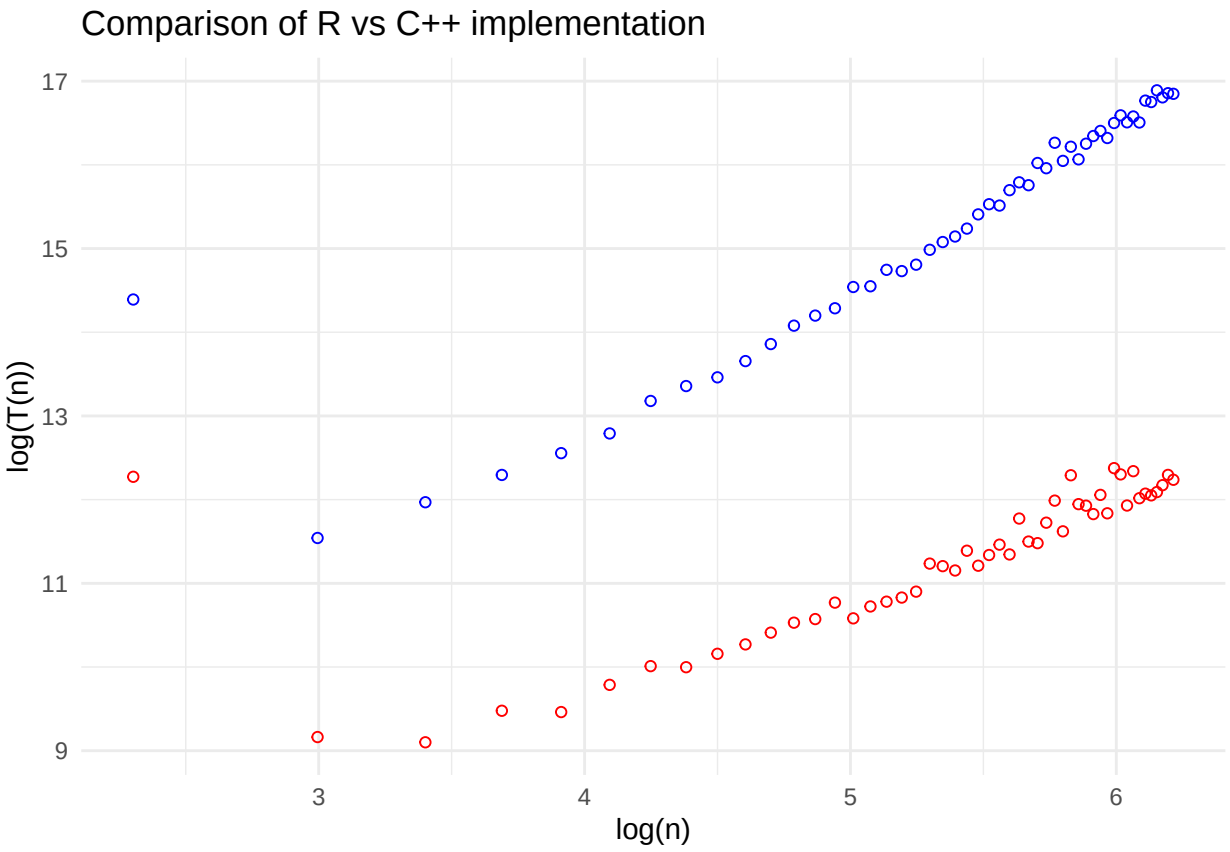
M2 Data Science, 2023 - 2024

```
library(microbenchmark)
library(Rcpp)
library(ggplot2)
```

Comparaison des temps d'exécution

```
source("StorageOptimisation/R/ffd_bin_packing.R")
sourceCpp("StorageOptimisation/src/ffdBinPacking.cpp")

measure_time_R <- function(n) {
  items <- sample(1:10, n, replace = TRUE)
  bin_size <- 10
  microbenchmark(ffd_bin_packing(items, bin_size), times = 5)
}

measure_time_Cpp <- function(n) {
  items <- sample(1:10, n, replace = TRUE)
  bin_size <- 10
  microbenchmark(ffd_bin_packing_Rcpp(items, bin_size), times = 5)
}

# Générer des tailles d'entrée (se concentrer sur les plus petites valeurs)
sizes <- seq(10, 500, by = 10)

# Mesurer le temps d'exécution pour l'implémentation en R
times_R <- sapply(sizes, function(n) mean(measure_time_R(n)$time))

# Mesurer le temps d'exécution pour l'implémentation en C++ (utiliser les mêmes tailles d'entrée)
times_Cpp <- sapply(sizes, function(n) mean(measure_time_Cpp(n)$time))
```

```
# Tracer le graphique log-log
ggplot() +
  geom_point(aes(x = log(sizes), y = log(times_R)), color = "blue", shape = 1) +
  geom_point(aes(x = log(sizes), y = log(times_Cpp)), color = "red", shape = 1) +
  labs(x = "log(n)", y = "log(T(n))", title = "Comparison of R vs C++ implementation") +
  theme_minimal()
```


Comparison of R vs C++ implementation
log(T(n))
log(n)
17
15
13
11
9
3
4
5
6