Intermediary

2024-07-18

```
library(readxl) # to read the initial file
library(magrittr) # for pipes
library(tibble) # for add_column
```

Preliminaries

```
# Importing the database with the artificial forest
trees <- read_excel("~/work/Sim_article2024/Data/artificial_forest_round.xls")
```

```
# Functions to produce a local variable
source("~/work/Sim_article2024/Useful_functions/Circle.R")
```

```
# Sample size
n <- 1000
```

Monte Carlo alone

10 000 Monte Carlo iterations are appropriated.

```
### Function to automate Monte Carlo method, to produce a variance

MonteCarlo <- function(n, B, f){
  # n: sample size;
  # B: number of iterations for MC;
  # f: function to consider (with 2 variables, x and y)

  p <- sum(rep((1/1000)*(1/1000), n)) # inclusion density
  est_MC <- c() # Vector containing all estimations
  for (b in 2:B) { # All in one "for" loop
    # Uniform sampling over the whole territory
    a1 <- runif(n, 0, 1000)
    a2 <- runif(n, 0, 1000)
    # Variable of interest derived from the sample
    a <- mapply(f, a1, a2)
    est_MC <- append(est_MC, sum(a/p))
  }

  return(var(est_MC))
  # Monte-Carlo variance
}
```

Bootstrap for each Monte Carlo iteration

1 000 Monte Carlo iterations and 400 bootstrap iterations for each one are appropriated.

```r
f_boot <- function(df_sample, B){
  # df_sample: original sample;
  # B: number of bootstrap samples made thanks to df_sample

  df_boot <- tibble(.rows = (nrow(df_sample)-1))
  for (b in 1:B) {
    nb <- sample(1:nrow(df_sample), (nrow(df_sample)-1), replace = TRUE)
    # (n-1) points at random
    sample_b <- df_sample[nb,]
    df_boot <- df_boot %>% add_column(new_col = sample_b)
    colnames(df_boot)[b] <- paste("ech", b, sep = "")
  }
  return(do.call(data.frame, df_boot))
  # Return a data frame with B samples of 2 variables and of size n
}
```

```r
# Making of one estimation for each sample made by the bootstrap method
est_boot <- function(df, p){
  # df: dataframe with 3*B columns and n rows

  n <- nrow(df) + 1 # sample size of the initial sample
  nboot <- length(df)/3 # number of bootstrap samples
  vec_est <- c()
  for (c in 1:nboot) {
    vec_est <- append(vec_est, (n/(n-1))*sum(df[,3*c]/p))
    # (n/(n-1)) is the correction needed since we choose only (n-1) points
  }
  vec_est <- sort(vec_est)
  return(vec_est)
  # Vector with one estimation for each sample
}
```

```r
### Function to automate Monte Carlo + bootstrap method

MC_boot <- function(n, B1, B2, f){
  # n: sample size; B1: number of iterations for MC;
  # B2: number of iterations for bootstrap;
  # f: function to consider (2 arguments)

  p <- sum(rep((1/1000)*(1/1000), n))
  var_est_boot <- c()
  for (b in 2:B1) {
    # One MC iteration at a time
    random_points <- data.frame(x = runif(n, 0, 1000), y = runif(n, 0, 1000))
    random_points$var <- mapply(f, random_points[, 1], random_points[, 2])
    a <- sum(random_points$var/p)

    # Bootstrap iterations
    df_boot <- f_boot(random_points, B2)
    est_boot_val <- est_boot(df_boot, p)
    var_est_boot <- append(var_est_boot, var(est_boot_val))
  }
  return(mean(var_est_boot))
  # Mean of the bootstrap variance
```

```
}
```

Second example: volume of the forest

```
start.time <- Sys.time()

res1_vol <- MonteCarlo(n, 10000, circle_vol)
# Just one circle to begin with

end.time <- Sys.time()
time.taken <- end.time - start.time

time.taken
```

```
## Time difference of 2.246892 hours
```

```
cat("Monte Carlo variance:", res1_vol)
```

```
## Monte Carlo variance: 98332.87
```

```
# Ex:
# Monte Carlo variance:
```

```
start.time <- Sys.time()

res2_vol <- MC_boot(n, B1 = 1000, B2 = 400, f = circle_vol)
# Just one circle to begin with

end.time <- Sys.time()
time.taken <- end.time - start.time

time.taken
```

```
## Time difference of 23.60139 mins
```

```
cat("Mean of the variance of bootstrap samples:", res2_vol)
```

```
## Mean of the variance of bootstrap samples: 98594.44
```

```
# Ex:
# Mean of the variance of bootstrap samples:
```

```
cat("We get a difference of: ", 100*(max(res1_vol,res2_vol)-min(res1_vol,res2_vol))/min(res1_vol,res2_v
```

```
## We get a difference of: 0.2660091%
```

```
# Ex:
# We get a difference of:
```