Homework 7

Chad Huntebrinker

2024-10-10

Question 1 A function is defined in the following chunk to calculate the square of each element using a for loop. Convert this function into a vectorized function

# Define a function to calculate the square of each element using a for loop  
square\_with\_loop <- function(x) {  
 result <- numeric(length(x)) # Initialize an empty numeric vector to store results  
 for (i in seq\_along(x)) {  
 result[i] <- x[i]^2 # Calculate square of each element and store in result vector  
 }  
 return(result)  
}  
  
# Test the function  
numbers <- c(1, 2, 3, 4, 5)  
print(square\_with\_loop(numbers))

## [1] 1 4 9 16 25

Answer to Question 1

square\_with\_loop <- function(x) {  
 return(x^2)  
}  
  
# Test the function  
numbers <- c(1, 2, 3, 4, 5)  
print(square\_with\_loop(numbers))

## [1] 1 4 9 16 25

Question 2 A function is defined in the following chunk to calculate the cumulative sum of elements in a given numeric vector using a for loop.Convert this function into a vectorized function

# Define function to calculate cumulative sum using a for loop  
cumulative\_sum\_for\_loop <- function(x) {  
 result <- numeric(length(x)) # Initialize a numeric vector to store cumulative sum  
 sum\_so\_far <- 0 # Initialize variable to keep track of sum so far  
 for (i in seq\_along(x)) {  
 sum\_so\_far <- sum\_so\_far + x[i] # Update sum so far  
 result[i] <- sum\_so\_far # Store cumulative sum in result vector  
 }  
 return(result)  
}  
  
# Test the function  
numbers <- c(1, 2, 3, 4, 5)  
print(cumulative\_sum\_for\_loop(numbers))

## [1] 1 3 6 10 15

Answer to Question 2

# Define function to calculate cumulative sum using a for loop  
cumulative\_sum\_for\_loop <- function(x) {  
 return(sum(x))  
}  
  
# Test the function  
numbers <- c(1, 2, 3, 4, 5)  
print(cumulative\_sum\_for\_loop(numbers))

## [1] 15

Question 3 Perform k-means clustering on the Iris dataset both serially and in parallel. Perform clustering on the first four columns of the Iris dataset, excluding the species (5th) column. report the computation times

library(datasets)  
library(parallel)  
data(iris)  
fx <- function(nstart){  
 # Perform k-means clustering on a data matrix Boston.  
 return(kmeans(iris\_data, centers = 4, nstart = nstart))  
}  
#Get the first four columns  
iris\_data <- data.frame(iris$Sepal.Length, iris$Sepal.Width, iris$Petal.Length, iris$Petal.Width)  
starts <- rep(100, 400)  
  
#Complete serieally  
system.time(results <- lapply(starts, fx))

## user system elapsed   
## 1.19 0.07 2.47

#Complete parallel  
numCores <- detectCores()  
cl <- makeCluster(8)  
clusterExport(cl=cl, c('iris\_data'))  
results <- parLapply(cl,starts, fx)  
system.time(results <- parLapply(cl,starts, fx))

## user system elapsed   
## 0.00 0.00 0.61

stopCluster(cl)