# Experiment 03: Optimizing Parking Allocation with VIP Area Using One-Dimensional Vectors in R

Name: Sparsh Karna

Registration Number: 23BDS1172

Course: Programming for Data Science Lab

Date: 31-07-25

# 1. Title of the Experiment

Optimizing Parking Allocation with VIP Area Using One-Dimensional Vectors in R

# 2. Objective

To simulate a real-world parking system using one-dimensional vectors in R and optimize the allocation of different types of vehicles, including VIPs, using only vector operations and user-defined functions.

# 3. Problem Statement

Simulate the vehicle parking system at VIT Chennai consisting of 5 bike slots, 4 car slots, and a fixed VIP parking slot using only one-dimensional vectors. Allocate the arriving vehicles to the appropriate slots optimally, ensuring VIPs are parked exclusively in the VIP zone. Generate a human-readable summary.

# 4. Logic / Algorithm Used

- Use sample() to generate random capacities and vehicle arrivals.

- Initialize separate vectors for capacity and usage for bikes and cars.

- VIP parking handled via a scalar.

- allocate\_parking() function loops over slots to allocate as many vehicles as possible.

- allocate\_vip() handles VIP parking with exact capacity check.

- display\_parking\_status() prints used and total capacities with full indicators.

- display\_utilization\_report() calculates and prints parking statistics.

# 5. Screenshots of Output

# 6. R Code (well-commented)

# Vector Initialization Section

# Random capacities for 5 bike parking slots (150–500)

bike\_capacity <- sample(150:500, 5, replace=TRUE)

bike\_used <- rep(0, 5) # Initialize usage to 0

# Random capacities for 4 car parking slots (150–500)

car\_capacity <- sample(150:500, 4, replace=TRUE)

car\_used <- rep(0, 4) # Initialize usage to 0

# VIP Parking has a fixed capacity of 10

vip\_capacity <- 10

vip\_used <- 0

# Simulate number of arriving vehicles

bikes\_arriving <- sample(500:800, 1)

cars\_arriving <- sample(400:600, 1)

vips\_arriving <- sample(5:15, 1)

# User-Defined Functions Section

#' @title: Allocate Vehicles

#' @description: Allocates incoming vehicles to available parking slots

#' @param arriving: number of incoming vehicles

#' @param capacity: vector of slot capacities

#' @param used: vector of used slots (modifiable)

#' @returns updated 'used' vector

allocate\_parking <- function(arriving, capacity, used) {

for (i in seq\_along(capacity)) {

available <- capacity[i] - used[i]

if (arriving > 0 && available > 0) {

to\_allocate <- min(arriving, available)

used[i] <- used[i] + to\_allocate

arriving <- arriving - to\_allocate

}

}

if (arriving > 0) {

cat("No more available space in this category.\n")

}

return(used)

}

#' @title: Allocate VIP Vehicles

#' @description: Allocates VIP vehicles to the VIP parking only

#' @param arriving: number of VIPs

#' @param capacity: fixed capacity (10)

#' @param used: current VIP usage (modifiable)

#' @returns updated 'used' value

allocate\_vip <- function(arriving, capacity, used) {

available <- capacity - used

to\_allocate <- min(arriving, available)

used <- used + to\_allocate

cat("Only", to\_allocate, "VIPs accommodated out of", arriving, "\n")

if (to\_allocate < arriving) {

cat("No more available space in this category.\n")

}

return(used)

}

#' @title: Display Parking Status

#' @description: Prints used vs total capacity of each parking slot

#' @param label: string label (BikeParking/CarParking)

#' @param capacity: vector of capacities

#' @param used: vector of used slots

display\_parking\_status <- function(label, capacity, used) {

cat(label, "Status:\n")

full\_slots <- c()

for (i in seq\_along(capacity)) {

status <- ifelse(used[i] == capacity[i], "[FULL]", "")

cat("Slot", i, "-> Used:", used[i], "/", capacity[i], status, "\n")

if (used[i] == capacity[i]) full\_slots <- c(full\_slots, i)

}

if (length(full\_slots) > 0) {

cat("\n", label, "FULL in slot(s):", paste(full\_slots, collapse=", "), "\n\n")

}

}

#' @title: Display Utilization Report

#' @description: Prints total, used, remaining and percentage utilization

#' @param total\_capacity: numeric vector of all capacities

#' @param total\_used: numeric vector of all usages

display\_utilization\_report <- function(total\_capacity, total\_used) {

total\_cap <- sum(total\_capacity)

used\_cap <- sum(total\_used)

remaining\_cap <- total\_cap - used\_cap

utilization <- round((used\_cap / total\_cap) \* 100, 2)

cat("=========== Parking Utilization Report ===========\n")

cat("Total Capacity:", total\_cap, "\n")

cat("Used Capacity:", used\_cap, "\n")

cat("Remaining Capacity:", remaining\_cap, "\n")

cat("Utilization:", utilization, "%\n")

}

# Allocation Execution Section

cat("=== Parking Allocation Simulation ===\n\n")

# Allocate VIPs

vip\_used <- allocate\_vip(vips\_arriving, vip\_capacity, vip\_used)

cat("\n")

# Allocate Bikes

bike\_used <- allocate\_parking(bikes\_arriving, bike\_capacity, bike\_used)

cat("\n")

# Allocate Cars

car\_used <- allocate\_parking(cars\_arriving, car\_capacity, car\_used)

cat("\n")

# Final Parking Status Summary

cat("=========== Final Parking Status ===========\n\n")

display\_parking\_status("BikeParking", bike\_capacity, bike\_used)

display\_parking\_status("CarParking", car\_capacity, car\_used)

cat("VIP Parking Used:", vip\_used, "/", vip\_capacity)

if (vip\_used == vip\_capacity) cat(" [FULL]\n") else cat("\n")

cat("\n")

# Utilization Report

all\_capacities <- c(bike\_capacity, car\_capacity, vip\_capacity)

all\_usages <- c(bike\_used, car\_used, vip\_used)

display\_utilization\_report(all\_capacities, all\_usages)

cat("\n=== Analysis Complete ===\n")

# 7. Sample Output

=== Parking Allocation Simulation ===

Only 10 VIPs accommodated out of 13

No more available space in this category.

=========== Final Parking Status ===========

BikeParking Status:

Slot 1 -> Used: 336 / 336 [FULL]

Slot 2 -> Used: 217 / 257

Slot 3 -> Used: 0 / 152

Slot 4 -> Used: 0 / 494

Slot 5 -> Used: 0 / 177

BikeParking FULL in slot(s): 1

CarParking Status:

Slot 1 -> Used: 320 / 320 [FULL]

Slot 2 -> Used: 243 / 243 [FULL]

Slot 3 -> Used: 8 / 342

Slot 4 -> Used: 0 / 384

CarParking FULL in slot(s): 1, 2

VIP Parking Used: 10 / 10 [FULL]

=========== Parking Utilization Report ===========

Total Capacity: 2715

Used Capacity: 1134

Remaining Capacity: 1581

Utilization: 41.77 %

=== Analysis Complete ===

# 8. Conclusion

The parking system was successfully simulated using only 1D vectors and modular user-defined functions. VIP parking was allocated exclusively, and the output clearly displayed full status and utilization. This lab reinforced the practical use of built-in functions and efficient allocation logic under resource constraints.