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

```
> source("~/dev/programming_for_data_science_23BDS1172/lab3/Analysis.R")
=== 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 ===
> -
```

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)
}
```

```r
#' @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.