**UNIT 1: Introduction to R Programming**

**1. Arithmetic Operations:**

a <- 8

b <- 2

sum <- a + b

print(sum)

diff <- a - b

print(diff)

product <- a \* b

print(product)

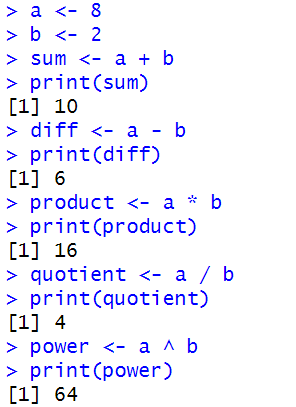
quotient <- a / b

print(quotient)

power <- a ^ b

print(power)

**Output:**

****

**2. Employee Details:**

employees <- data.frame(

ID = c(101, 102, 103),

Name = c("Alice", "Bob", "Charlie"),

Age = c(25, 30, 28),

Salary = c(50000, 60000, 55000)

)

employees$Department <- c("HR", "IT", "Finance")

high\_salary <- subset(employees, Salary > 55000)

sorted\_employees <- employees[order(employees$Salary), ]

print("Original Data Frame:")

print(employees)

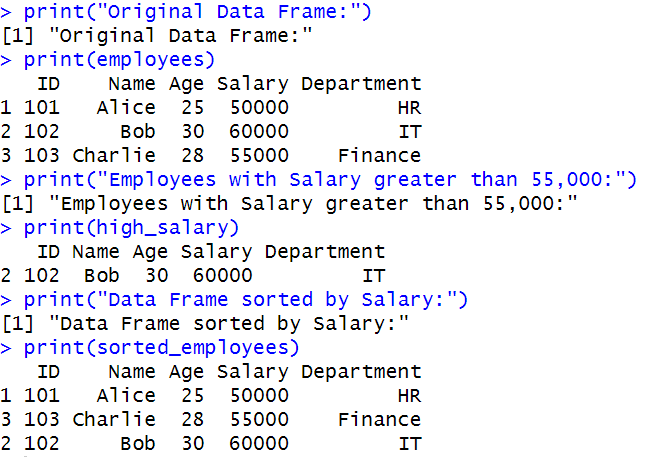
print("Employees with Salary greater than 55,000:")

print(high\_salary)

print("Data Frame sorted by Salary:")

print(sorted\_employees)

**Output:**

****

**3. Vector Sum:**

mat <- matrix(1:9, nrow = 3)

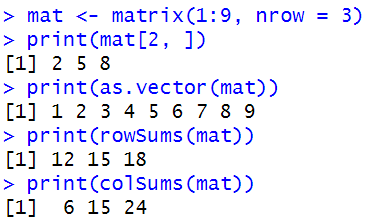
print(mat[2, ])

print(as.vector(mat))

print(rowSums(mat))

print(colSums(mat))

**Output:**

****

**4. Sequence of Numbers**

sequence <- 20:50

print("Sequence from 20 to 50:")

print(sequence)

mean\_value <- mean(20:60)

print("Mean of numbers from 20 to 60:")

print(mean\_value)

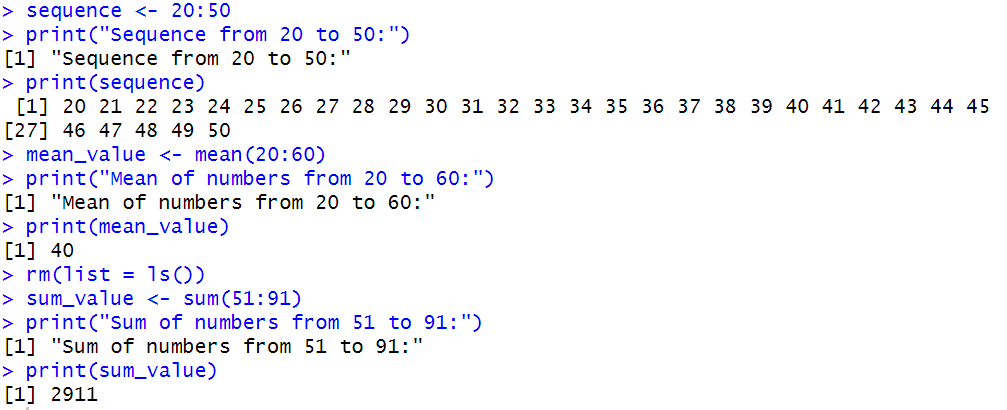
rm(list = ls())

sum\_value <- sum(51:91)

print("Sum of numbers from 51 to 91:")

print(sum\_value)

**Output:**

****

**5. English Letters**first\_10\_lower <- letters[1:10]

print("First 10 English letters in lowercase:")

print(first\_10\_lower)

last\_10\_upper <- LETTERS[17:26]

print("Last 10 English letters in uppercase:")

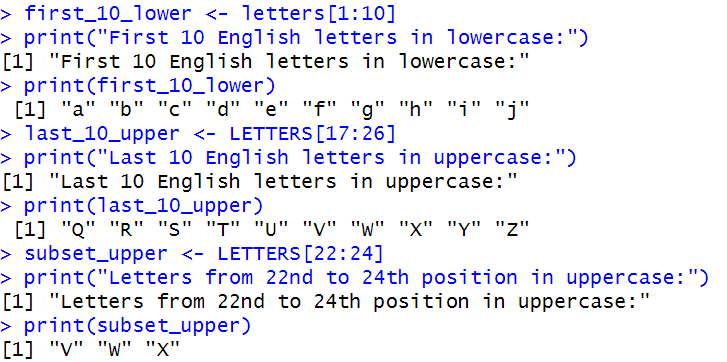
print(last\_10\_upper)

subset\_upper <- LETTERS[22:24]

print("Letters from 22nd to 24th position in uppercase:")

print(subset\_upper)

**Output:**

****

**6. Logical Operations**

numbers <- c(5, 12, 18, 25, 30, 35, 42, 50, 60, 75)

logical\_vector <- (numbers > 20 & numbers %% 2 == 0) | (numbers < 10)

print("Logical vector (Numbers > 20 AND even) OR (Numbers < 10):")

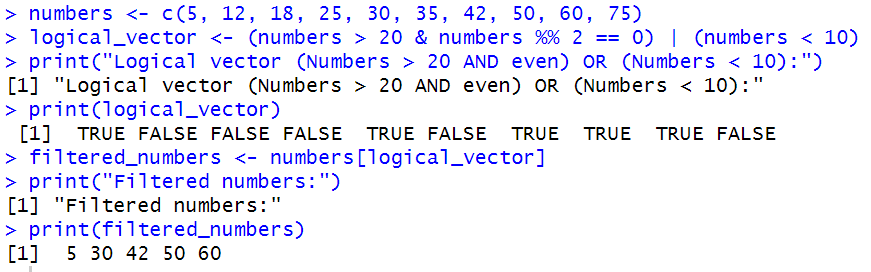
print(logical\_vector)

filtered\_numbers <- numbers[logical\_vector]

print("Filtered numbers:")

print(filtered\_numbers)

**Output:**

****

**7. Character Vector of Categorical Data**

colors <- c("Red", "Blue", "Green", "Blue", "Red", "Green", "Yellow")

color\_factor <- factor(colors)

print("Factor levels before modification:")

print(levels(color\_factor))

levels(color\_factor) <- c("Crimson", "SkyBlue", "Green", "Yellow")

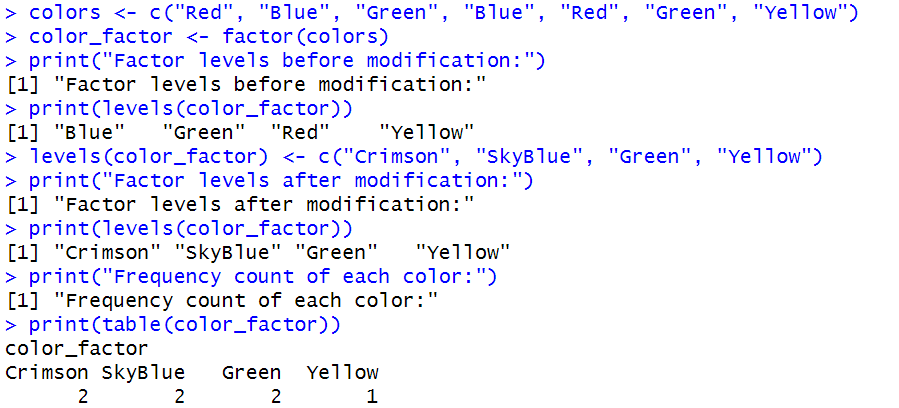
print("Factor levels after modification:")

print(levels(color\_factor))

print("Frequency count of each color:")

print(table(color\_factor))

**Output:**

****

**8. Data Types in R**

num\_val <- 12.5

int\_val <- 10L

char\_val <- "100"

log\_val <- TRUE

comp\_val <- 2 + 3i

num\_to\_int <- as.integer(num\_val)

char\_to\_num <- as.numeric(char\_val)

log\_to\_num <- as.numeric(log\_val)

int\_to\_char <- as.character(int\_val)

num\_to\_comp <- as.complex(num\_val)

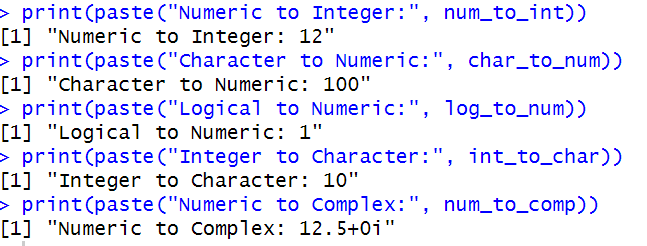
print(paste("Numeric to Integer:", num\_to\_int))

print(paste("Character to Numeric:", char\_to\_num))

print(paste("Logical to Numeric:", log\_to\_num))

print(paste("Integer to Character:", int\_to\_char))

print(paste("Numeric to Complex:", num\_to\_comp))  
**Output:**

****

**9. Matrix**

mat1 <- matrix(1:20, nrow = 5, ncol = 4)

print("5x4 Matrix:")

print(mat1)

mat2 <- matrix(1:9, nrow = 3, byrow = TRUE,

dimnames = list(c("R1", "R2", "R3"), c("C1", "C2", "C3")))

print("3x3 Matrix (filled by rows):")

print(mat2)

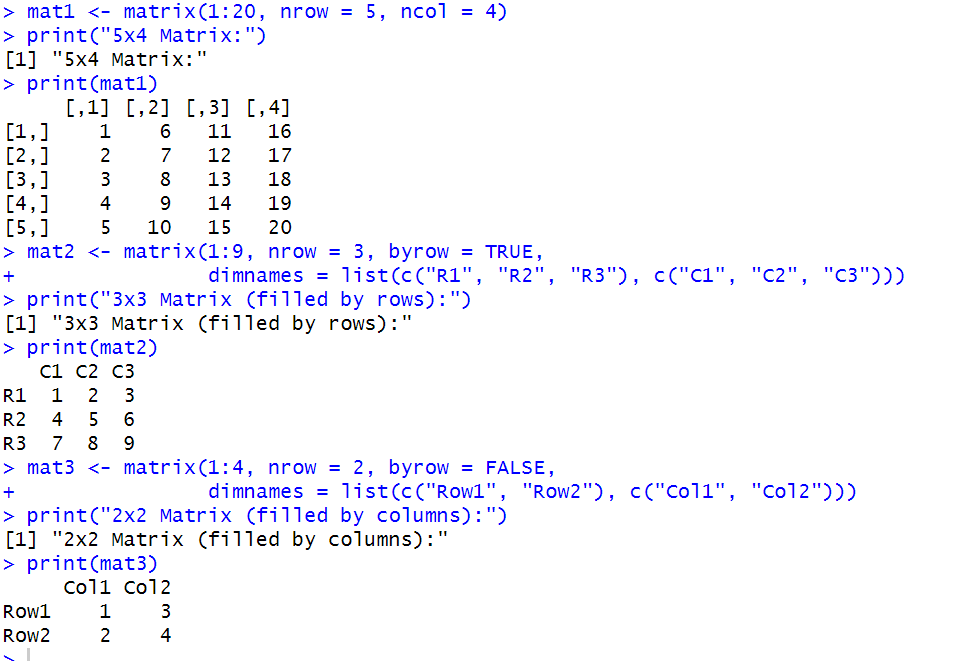
mat3 <- matrix(1:4, nrow = 2, byrow = FALSE,

dimnames = list(c("Row1", "Row2"), c("Col1", "Col2")))

print("2x2 Matrix (filled by columns):")

print(mat3)

**Output:**

****

**10. Two-dimensional 5x3 Array**

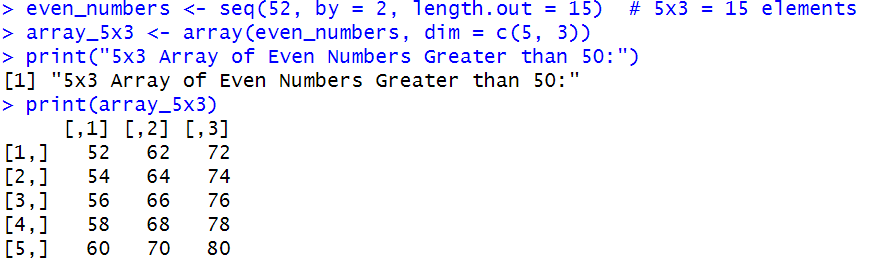
even\_numbers <- seq(52, by = 2, length.out = 15) # 5x3 = 15 elements

array\_5x3 <- array(even\_numbers, dim = c(5, 3))

print("5x3 Array of Even Numbers Greater than 50:")

print(array\_5x3)

**Output:**

****

**11. Access Values in a Vector**

numbers <- c(10, 20, 30, 40, 50)

print("Original Vector:")

print(numbers)

print("First element:")

print(numbers[1]) # First element

print("Last element:")

print(numbers[length(numbers)]) # Last element

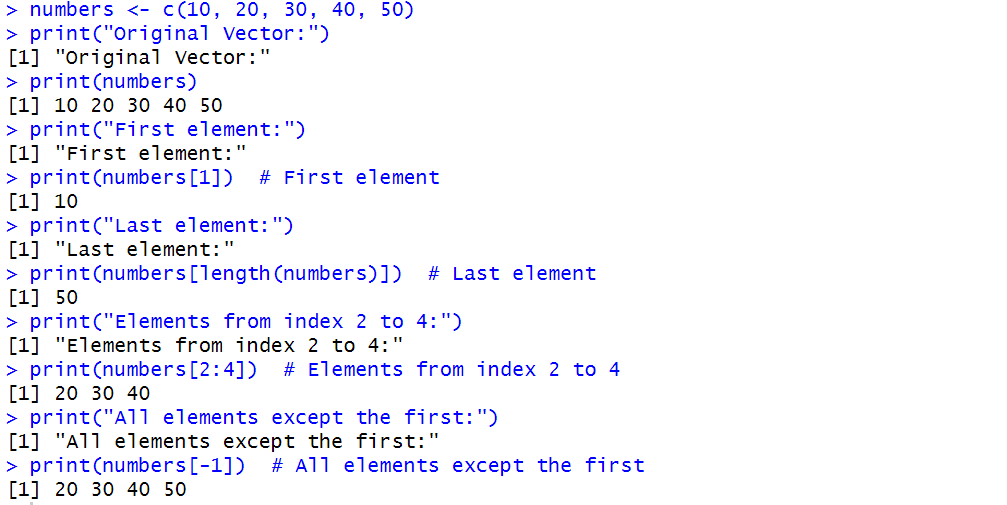
print("Elements from index 2 to 4:")

print(numbers[2:4]) # Elements from index 2 to 4

print("All elements except the first:")

print(numbers[-1]) # All elements except the first

**Output:**



**12. Nth Smallest Value in Vector**

nth\_smallest <- function(vec, n) {

sorted\_vec <- sort(vec)

return(sorted\_vec[n])

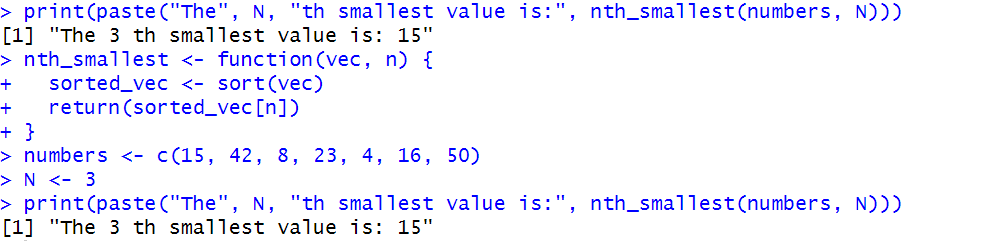
}

numbers <- c(15, 42, 8, 23, 4, 16, 50)

N <- 3

print(paste("The", N, "th smallest value is:", nth\_smallest(numbers, N)))

**Output:**

****

**13. Concatenate a Vector of Strings**

words <- c("Hello", "World", "from", "R")

concatenated\_string <- paste(words, collapse = " ")

print("Concatenated string with spaces:")

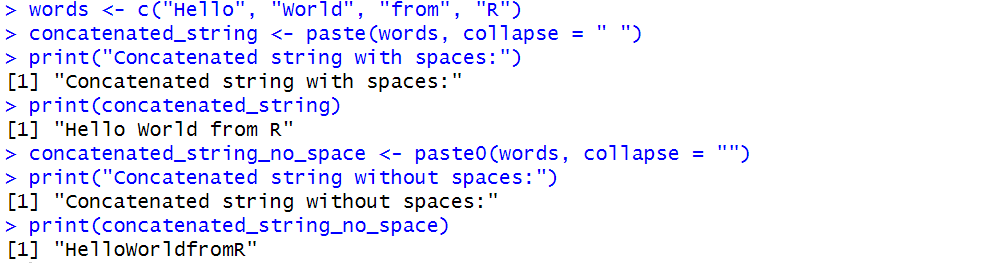
print(concatenated\_string)

concatenated\_string\_no\_space <- paste0(words, collapse = "")

print("Concatenated string without spaces:")

print(concatenated\_string\_no\_space)

**Output:**

****

**14. Row and Column Indices of the Maximum and Minimum Values**

mat <- matrix(c(10, 25, 7, 40, 15, 33, 2, 19, 50), nrow = 3, byrow = TRUE)

print("Matrix:")

print(mat)

max\_index <- which(mat == max(mat), arr.ind = TRUE)

min\_index <- which(mat == min(mat), arr.ind = TRUE)

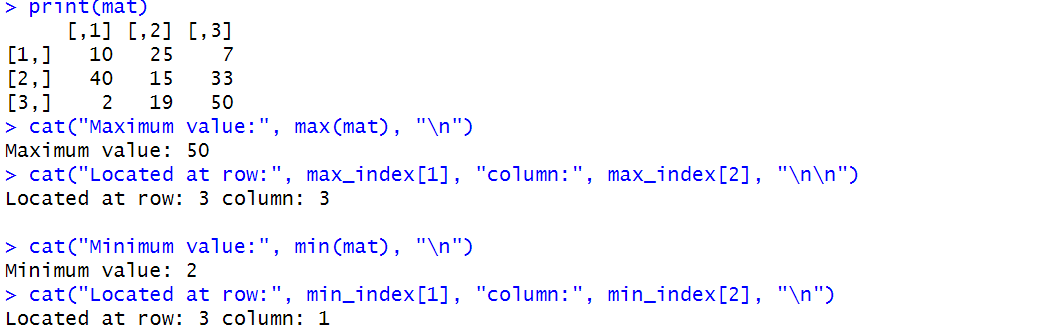
cat("Maximum value:", max(mat), "\n")

cat("Located at row:", max\_index[1], "column:", max\_index[2], "\n\n")

cat("Minimum value:", min(mat), "\n")

cat("Located at row:", min\_index[1], "column:", min\_index[2], "\n")

**Output:**

****

**15. FizzBuzz**

for (i in 1:100) {

if (i %% 3 == 0 && i %% 5 == 0) {

print("FizzBuzz")

} else if (i %% 3 == 0) {

print("Fizz")

} else if (i %% 5 == 0) {

print("Buzz")

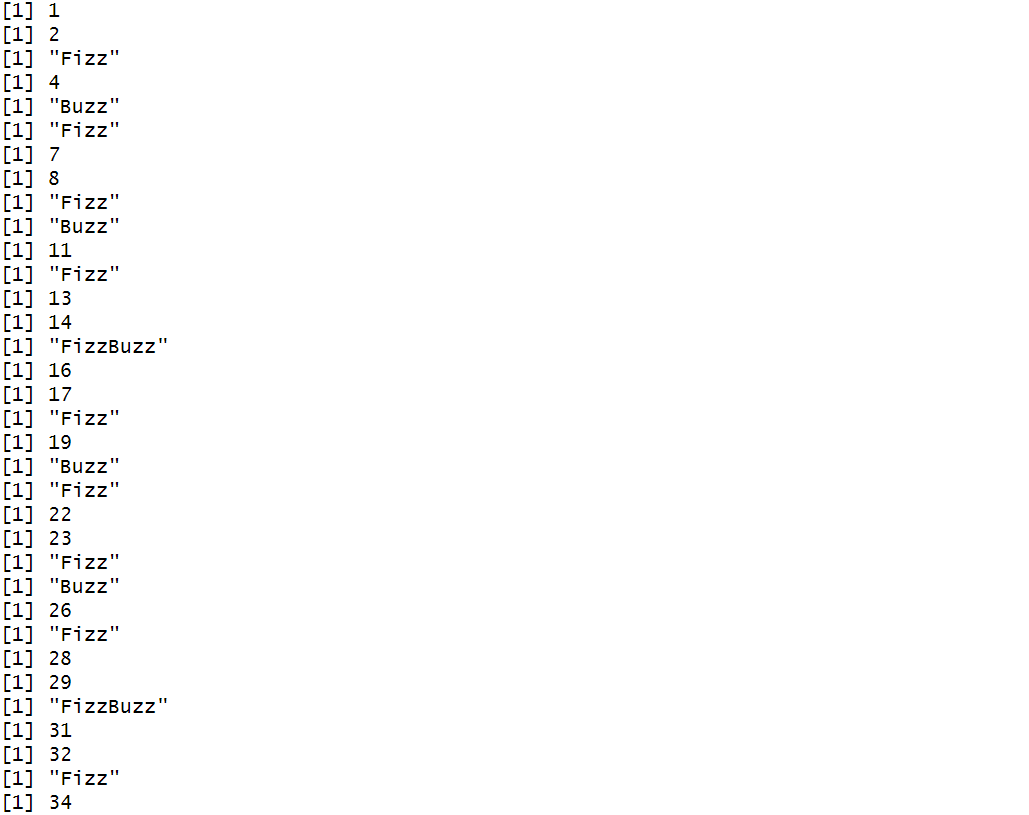
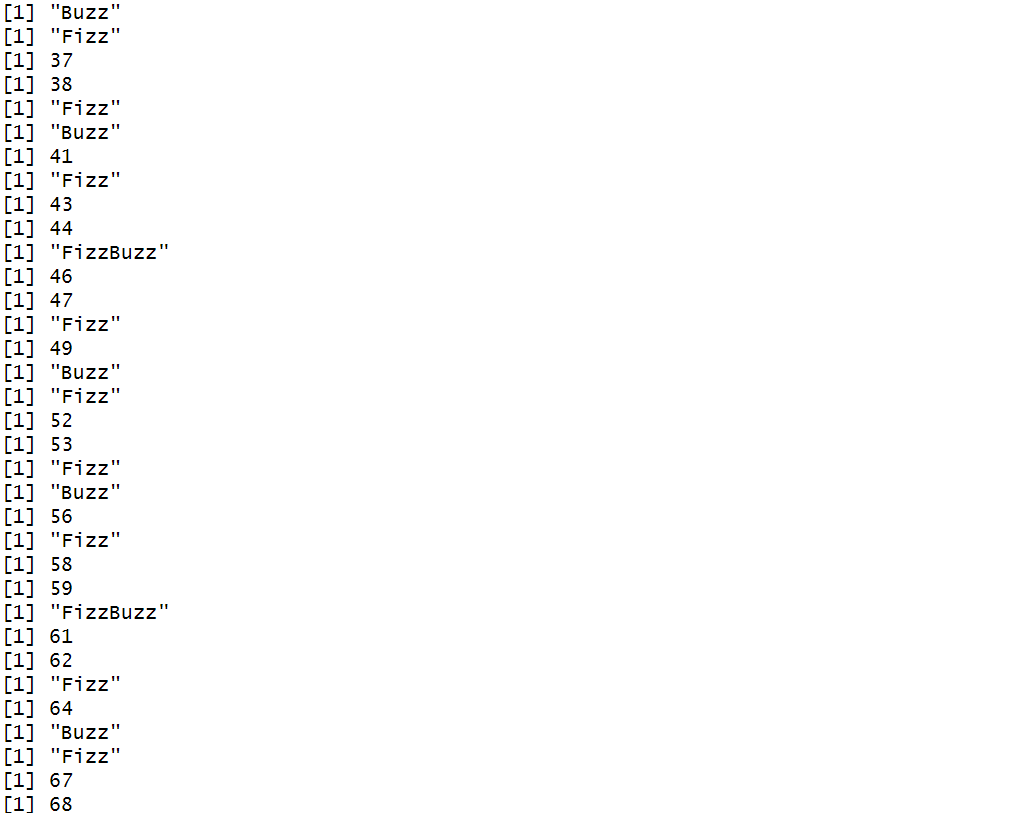
} else {

print(i)

}

}

**Output:**

**** ****

****

**16. Converts A List to a Data Frame with Specific Column Names**

my\_list <- list(

c("John", "Alice", "Bob"),

c(25, 30, 22),

c("M", "F", "M")

)

df <- data.frame(

Name = my\_list[[1]],

Age = my\_list[[2]],

Gender = my\_list[[3]],

stringsAsFactors = FALSE

)

print("Converted Data Frame:")

print(df)

**Output:**

****

**17. Summary of Employee Details**

employees <- data.frame(

EmployeeID = c(101, 102, 103, 104, 105),

Name = c("John", "Alice", "David", "Maria", "Tom"),

Age = c(28, 34, 26, 31, 29),

Department = c("HR", "Finance", "IT", "Marketing", "IT"),

Salary = c(50000, 60000, 55000, 58000, 53000),

stringsAsFactors = FALSE

)

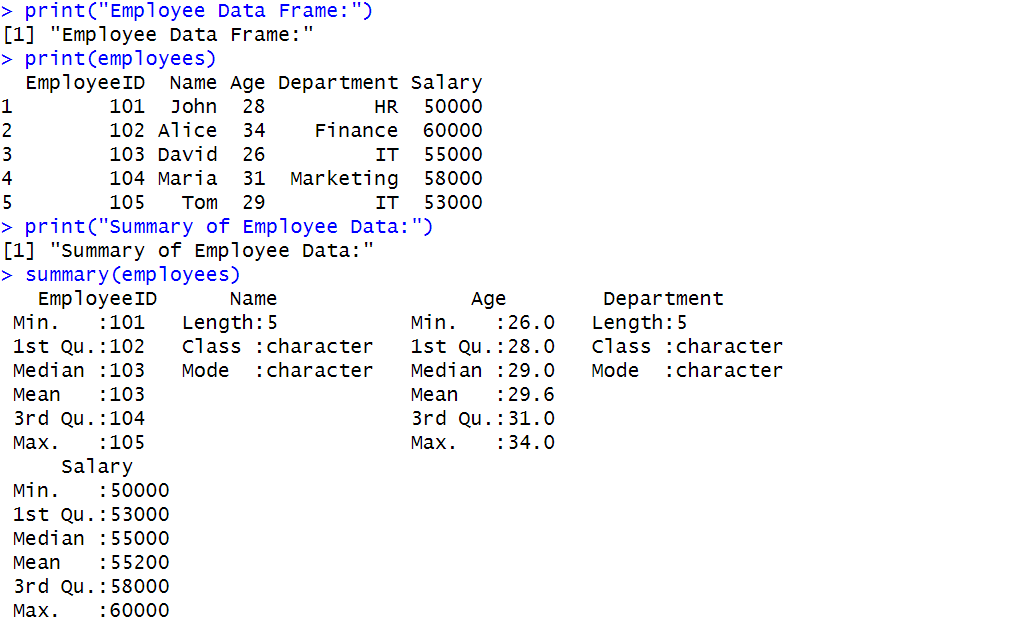
print("Employee Data Frame:")

print(employees)

print("Summary of Employee Data:")

summary(employees)

**Output:**

****

**18. Minimum and Maximum of Vector**

vec <- c(12, 45, 7, 89, 23, 56, 3, 78)

print("Given Vector:")

print(vec)

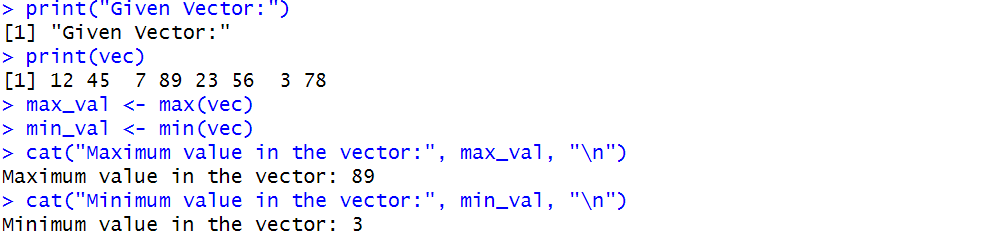
max\_val <- max(vec)

min\_val <- min(vec)

cat("Maximum value in the vector:", max\_val, "\n")

cat("Minimum value in the vector:", min\_val, "\n")

**Output:**

****

**19. 3x3x2 Array**

vec1 <- c(1, 2, 3, 4, 5, 6)

vec2 <- c(7, 8, 9, 10, 11, 12)

combined\_vec <- c(vec1, vec2)

my\_array <- array(combined\_vec, dim = c(3, 3, 2))

print("3x3x2 Array:")

print(my\_array)

**Output:**

****

**20. Assign grades**

score <- 85

if (score >= 90) {

grade <- "A"

} else if (score >= 80) {

grade <- "B"

} else if (score >= 70) {

grade <- "C"

} else if (score >= 60) {

grade <- "D"

} else {

grade <- "F"

}

cat("Score:", score, "\n")

cat("Grade assigned:", grade, "\n")

**Output:**

****