

# ITA04- STATISTICS WITH R PROGRAMMING

1. Write a R program to create an array of two 3x3 matrices each with 3 rows and 3 columns from two given two vectors. Print the second row of the second matrix of the array and the element in the 3rd row and 3rd column of the 1st matrix

The screenshot shows the R Studio interface with two windows open:

- R Console:** Displays the output of the R code. It shows the creation of a 3x3 matrix, the extraction of the second row, and the access of a specific element in the first matrix.
- R Script:** Shows the R code being executed, including comments and function calls.

**R Console Output:**

```

Row1  1  4  7
Row2  2  5  8
Row3  3  6  9

, , Matrix2

      Col1 Col2 Col3
Row1  10  13  16
Row2  11  14  17
Row3  12  15  18

>
> cat("\nSecond row of the second Matrix:\n")
Second row of the second matrix:
> print(your_array["Row2", , "Matrix2"])
Col1 Col2 Col3
 11  14  17
>
> cat("\nElement in the 3rd row and 3rd column of the 1st matrix:\n")
Element in the 3rd row and 3rd column of the 1st matrix:
> print(your_array["Row3", "Col3", "Matrix1"])
[1] 9
> ]

```

**R Script Code:**

```

# C:\Users\Kevin Allen\OneDrive\Desktop\R Programs\DAY-03\01 the second row of the second ...
vector1 <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)
vector2 <- c(10, 11, 12, 13, 14, 15, 16, 17, 18)
your_array <- array(c(vector1, vector2), dim = c(3, 3, 2), dimnames = list(c("Row
print(your_array)

cat("\nSecond row of the second matrix:\n")
print(your_array["Row2", , "Matrix2"])

cat("\nElement in the 3rd row and 3rd column of the 1st matrix:\n")
print(your_array["Row3", "Col3", "Matrix1"])

```

2. Write a R program to combine three arrays so that the first row of the first array is followed by the first row of the second array and then first row of the third array

The screenshot displays two instances of the R Console window from R Studio. The left window contains the following R code:

```
> combined_array <- rbind(array1, array2, array3)
> print(combined_array)
```

The output of the second command is displayed as a matrix:

```
[,1] [,2] [,3]
[1,] 1 2 3
[2,] 4 5 6
[3,] 7 8 9
```

The right window shows the same code being entered or executed, with the title bar indicating the file path: C:\Users\Kevin Allen\OneDrive\Desktop\R Programs\DAY-03\02]. The code in the right window is identical to the one in the left window.

3. Write a R program to create an array using four given columns, three given rows, and two given tables and display the content of the array

```

File Edit Packages Windows Help
R Console
>
>
>
> data_table1 <- matrix(1:12, nrow = 3)
> data_table2 <- matrix(13:24, nrow = 3)
> your_array <- array(c(data_table1, data_table2), dim = c(3, 4, 2), dimnames = list(
>
> print(your_array)
, , Table1
   Col1 Col2 Col3 Col4
Row1  1    4    7   10
Row2  2    5    8   11
Row3  3    6    9   12
, , Table2
   Col1 Col2 Col3 Col4
Row1 13   16   19   22
Row2 14   17   20   23
Row3 15   18   21   24
>

```

```

C:\Users\Kevin Allen\OneDrive\Desktop\R Programs\DAT-03\03 sequence of even integers grea...
data_table1 <- matrix(1:12, nrow = 3)
data_table2 <- matrix(13:24, nrow = 3)
your_array <- array(c(data_table1, data_table2), dim = c(3, 4, 2), dimnames = list(
print(your_array)

```

4. Write a R program to create a two-dimensional 5x3 array of sequence of even integers greater than 50

```

File Edit Packages Windows Help
R Console
> print(your_array)
, , Table1
   Col1 Col2 Col3 Col4
Row1  1    4    7   10
Row2  2    5    8   11
Row3  3    6    9   12
, , Table2
   Col1 Col2 Col3 Col4
Row1 13   16   19   22
Row2 14   17   20   23
Row3 15   18   21   24
>
> your_array <- matrix(seq(from = 52, by = 2, length.out = 5 * 3), nrow = 5, ncol =
>
> print(your_array)
[,1] [,2] [,3]
[1,] 52 62 72
[2,] 54 64 74
[3,] 56 66 76
[4,] 58 68 78
[5,] 60 70 80
>

```

```

C:\Users\Kevin Allen\OneDrive\Desktop\R Programs\DAY-03\04 R - R Editor
your_array <- matrix(seq(from = 52, by = 2, length.out = 5 * 3), nrow = 5, ncol =
print(your_array)

```

5. Create below data frame exam\_data = data.frame(name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'), score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19), attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1), qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')) a. Write a R program to extract 3rd and 5th rows with 1st and 3rd columns from a given data frame b. Write a R program to add a new column named country in a given data frame Country<-c("USA", "USA", "USA", "USA", "UK", "USA", "USA", "India", "USA", "USA")

```

# R Console
3 Katherine 2
5 Emily 2
> exam_data <- data.frame(
+ name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matth
+ score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
+ attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
+ qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
+ )
>
> selected_data <- exam_data[c(3, 5), c(1, 3)]
>
> print(selected_data)
  name attempts
3 Katherine 2
5 Emily 2
> exam_data <- data.frame(
+ name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matth
+ score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
+ attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
+ qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
+ )
>
> country <- c("USA", "USA", "USA", "USA", "UK", "USA", "USA", "India", "USA", "U
> exam_data$country <- country
>
> print(exam_data)
  name score attempts qualify country
1 Anastasia 12.5 1 yes USA
2 Dima 9.0 3 no USA
3 Katherine 16.5 2 yes USA
4 James 12.0 3 no USA
5 Emily 9.0 2 no UK
6 Michael 20.0 3 yes USA
7 Matthew 14.5 1 yes USA
8 Laura 13.5 1 no India
9 Kevin 8.0 2 no USA
10 Jonas 19.0 1 yes USA
>

# C:\Users\Kevin Allen\OneDrive\Desktop\R Programs\DAY-03\05 extract 3rd and 5th rows with...
exam_data <- data.frame(
name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matth
score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)
selected_data <- exam_data[c(3, 5), c(1, 3)]
print(selected_data)
exam_data <- data.frame(
name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matth
score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)
country <- c("USA", "USA", "USA", "USA", "UK", "USA", "USA", "India", "USA", "USA
exam_data$country <- country
print(exam_data)

```

6. Write a R program to add new row(s) to an existing data frame new\_exam\_data = data.frame(name = c('Robert', 'Sophia'), score = c(10.5, 9), attempts = c(1, 3), qualify = c('yes', 'no')) d. Write a R program to sort a given data frame by name and score e. Write a R program to save the information of a data frame in a file and display the information of the file.

```

# R Console
4 Michael 20.0 3 yes
7 Matthew 14.5 1 yes
8 Laura 13.5 1 no
9 Kevin 8.0 2 no
10 Jonas 19.0 1 yes
11 Robert 10.5 1 yes
12 Sophia 9.0 3 no
> sorted_data <- combined_data[order(combined_data$name, combined_data$score), ]
> print(sorted_data)
  name score attempts qualify
1 Anastasia 12.5 1 yes
2 Dima 9.0 3 no
3 Emily 9.0 2 no
4 James 12.0 3 no
5 Jonas 19.0 1 yes
6 Katherine 16.5 2 yes
7 Kevin 8.0 2 no
8 Laura 13.5 1 no
9 Matthew 14.5 1 yes
10 Michael 20.0 3 yes
11 Robert 10.5 1 yes
12 Sophia 9.0 3 no
> write.csv(sorted_data, "sorted_data.csv", row.names = FALSE)
>
> file_content <- read.csv("sorted_data.csv")
> print(file_content)
  name score attempts qualify
1 Anastasia 12.5 1 yes
2 Dima 9.0 3 no
3 Emily 9.0 2 no
4 James 12.0 3 no
5 Jonas 19.0 1 yes
6 Katherine 16.5 2 yes
7 Kevin 8.0 2 no
8 Laura 13.5 1 no
9 Matthew 14.5 1 yes
10 Michael 20.0 3 yes
11 Robert 10.5 1 yes
12 Sophia 9.0 3 no
>

# C:\Users\Kevin Allen\OneDrive\Desktop\R Programs\DAY-03\06 data frame or not Order the e...
exam_data <- data.frame(
name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matth
score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)
new_exam_data <- data.frame(name = c('Robert', 'Sophia'), score = c(10.5, 9), att
combined_data <- rbind(exam_data, new_exam_data)
print(combined_data)
sorted_data <- combined_data[order(combined_data$name, combined_data$score), ]
print(sorted_data)
write.csv(sorted_data, "sorted_data.csv", row.names = FALSE)
file_content <- read.csv("sorted_data.csv")
print(file_content)

```

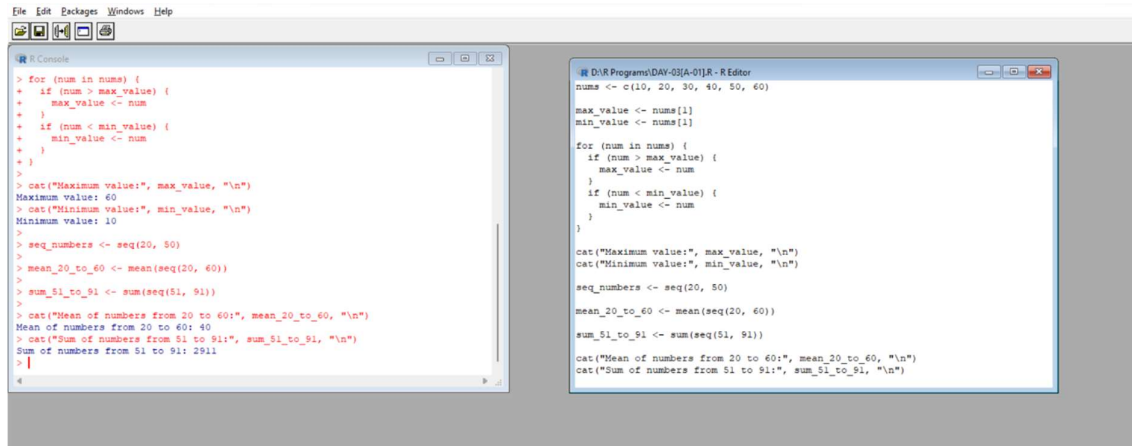
7. Write a R program to call the (built-in) dataset airquality. Check whether it is a data frame or not? Order the entire data frame by the first and second column. remove the variables 'Solar.R' and 'Wind' and display the data frame





# ANALYTICAL QUESTIONS

1. Write R Program to find maximum and minimum value of a given vector using control statement. `nums = c(10, 20, 30, 40, 50, 60)`
- a) Write a R program to create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of numbers from 51 to 91.

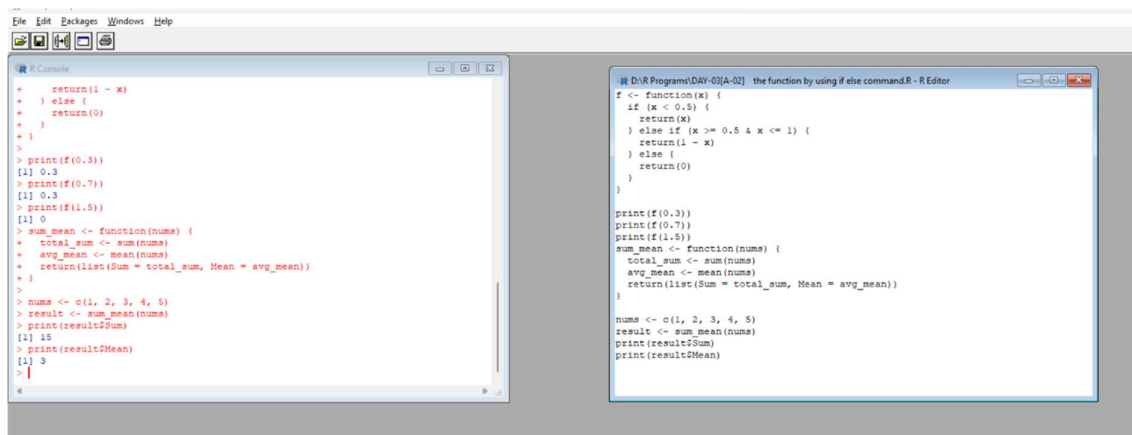


The screenshot shows two windows from an R environment. The left window is the R Console, and the right window is the R Editor. Both contain R code to find the maximum and minimum values of a vector and calculate the mean and sum of specific ranges.

```
R Console:
> for (num in nums) {
+   if (num > max_value) {
+     max_value <- num
+   }
+   if (num < min_value) {
+     min_value <- num
+   }
+ }
>
> cat("Maximum value:", max_value, "\n")
Maximum value: 60
> cat("Minimum value:", min_value, "\n")
Minimum value: 10
>
> seq_numbers <- seq(20, 50)
>
> mean_20_to_60 <- mean(seq(20, 60))
>
> sum_51_to_91 <- sum(seq(51, 91))
>
> cat("Mean of numbers from 20 to 60:", mean_20_to_60, "\n")
Mean of numbers from 20 to 60: 40
> cat("Sum of numbers from 51 to 91:", sum_51_to_91, "\n")
Sum of numbers from 51 to 91: 2911
> |

R Editor:
nums <- c(10, 20, 30, 40, 50, 60)
max_value <- nums[1]
min_value <- nums[1]
for (num in nums) {
  if (num > max_value) {
    max_value <- num
  }
  if (num < min_value) {
    min_value <- num
  }
}
cat("Maximum value:", max_value, "\n")
cat("Minimum value:", min_value, "\n")
seq_numbers <- seq(20, 50)
mean_20_to_60 <- mean(seq(20, 60))
sum_51_to_91 <- sum(seq(51, 91))
cat("Mean of numbers from 20 to 60:", mean_20_to_60, "\n")
cat("Sum of numbers from 51 to 91:", sum_51_to_91, "\n")
```

2. a) Write R code to the function by using if else command
- $$f(x) = \begin{cases} x & \text{if } x < 0.5 \\ (1-x) & \text{if } 0.5 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$
- (b) Write R function to find sum and mean of the given vector



The screenshot shows two windows from an R environment. The left window is the R Console, and the right window is the R Editor. Both contain R code to define a function `f(x)` and a function to calculate the sum and mean of a vector.

```
R Console:
> return(1 - x)
+ } else {
+   return(0)
+ }
+ }
>
> print(f(0.3))
[1] 0.3
> print(f(0.7))
[1] 0.3
> print(f(1.5))
[1] 0
>
> sum_mean <- function(nums) {
+   total_sum <- sum(nums)
+   avg_mean <- mean(nums)
+   return(list(Sum = total_sum, Mean = avg_mean))
+ }
>
> nums <- c(1, 2, 3, 4, 5)
> result <- sum_mean(nums)
> print(result$Sum)
[1] 15
> print(result$Mean)
[1] 3
> |

R Editor:
f <- function(x) {
  if (x < 0.5) {
    return(x)
  } else if (x >= 0.5 & x <= 1) {
    return(1 - x)
  } else {
    return(0)
  }
}

print(f(0.3))
print(f(0.7))
print(f(1.5))

sum_mean <- function(nums) {
  total_sum <- sum(nums)
  avg_mean <- mean(nums)
  return(list(Sum = total_sum, Mean = avg_mean))
}

nums <- c(1, 2, 3, 4, 5)
result <- sum_mean(nums)
print(result$Sum)
print(result$Mean)
```

3. What is data frame and how to create a data frame using the following data:

Height GPA

66 3.80

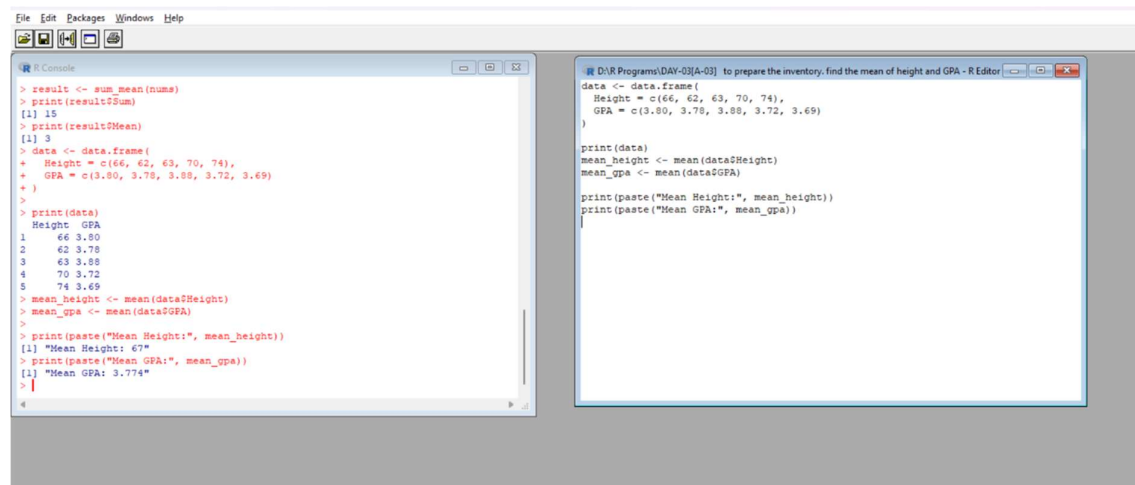
62 3.78

63 3.88

70 3.72

74 3.69

Write a R program to prepare the inventory. find the mean of height and GPA



The screenshot shows an R environment with two windows. The R Console window on the left displays the execution of R code, including the creation of a data frame 'data' with columns 'Height' and 'GPA', and the calculation of their means. The R Editor window on the right shows the source code for the program.

```
> result <- sum_mean(nums)
> print(result$sum)
[1] 15
> print(result$mean)
[1] 3
> data <- data.frame(
+   Height = c(66, 62, 63, 70, 74),
+   GPA = c(3.80, 3.78, 3.88, 3.72, 3.69)
+ )
> 
> print(data)
  Height GPA
1    66 3.80
2    62 3.78
3    63 3.88
4    70 3.72
5    74 3.69
> mean_height <- mean(data$Height)
> mean_gpa <- mean(data$GPA)
> 
> print(paste("Mean Height:", mean_height))
[1] "Mean Height: 67"
> print(paste("Mean GPA:", mean_gpa))
[1] "Mean GPA: 3.774"
> |
```

```
data <- data.frame(
  Height = c(66, 62, 63, 70, 74),
  GPA = c(3.80, 3.78, 3.88, 3.72, 3.69)
)

print(data)
mean_height <- mean(data$Height)
mean_gpa <- mean(data$GPA)

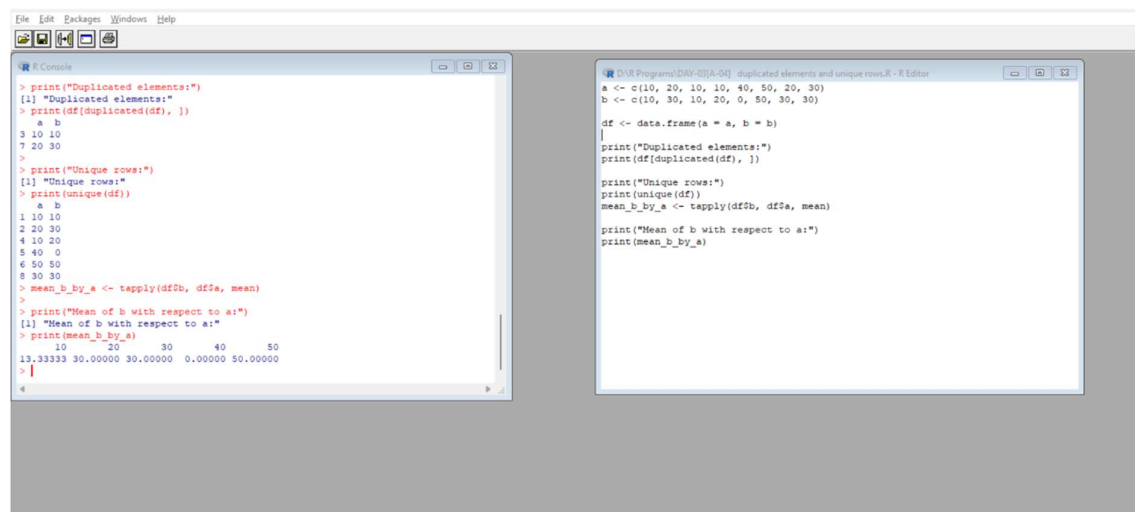
print(paste("Mean Height:", mean_height))
print(paste("Mean GPA:", mean_gpa))
```

4.(a)Write a R program to create a data frame using two given vectors and display the duplicated elements and unique rows of the said data frame. Sample data:

a = c (10,20,10,10,40,50,20,30)

b = c (10,30,10,20,0,50,30,30)

(b)write R code to find mean of “b” with respect to a in the above data frame "



The screenshot shows an R environment with two windows. The R Console window on the left displays the execution of R code, including the creation of a data frame 'df' with columns 'a' and 'b', and the calculation of the mean of 'b' with respect to 'a'. The R Editor window on the right shows the source code for the program.

```
> print("Duplicated elements:")
[1] "Duplicated elements:"
> print(df[duplicated(df), ])
  a b
3 10 10
7 20 30
> 
> print("Unique rows:")
[1] "Unique rows:"
> print(unique(df))
  a b
1 10 10
2 20 30
4 10 20
5 40 0
6 50 50
8 30 30
> mean_b_by_a <- tapply(df$b, df$a, mean)
> 
> print("Mean of b with respect to a:")
[1] "Mean of b with respect to a:"
> print(mean_b_by_a)
  10    20    30    40    50 
13.33333 30.00000 30.00000 0.00000 50.00000 
> |
```

```
a <- c(10, 20, 10, 10, 40, 50, 20, 30)
b <- c(10, 30, 10, 20, 0, 50, 30, 30)

df <- data.frame(a = a, b = b)

print("Duplicated elements:")
print(df[duplicated(df), ])

print("Unique rows:")
print(unique(df))

mean_b_by_a <- tapply(df$b, df$a, mean)

print("Mean of b with respect to a:")
print(mean_b_by_a)
```