**DAY 2 – LAB ASSESSMENT**

**ITA0443-STATISTICS WITH R PROGRAMMING**

**INDU SEKHAR**

**192124123**

**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.**

**CODE:**

# Create two vectors

vec1 <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)

vec2 <- c(10, 11, 12, 13, 14, 15, 16, 17, 18)

# Convert vectors to matrices

mat1 <- matrix(vec1, nrow = 3, ncol = 3)

mat2 <- matrix(vec2, nrow = 3, ncol = 3)

# Create an array of the two matrices

array\_of\_matrices <- array(c(mat1, mat2), dim = c(3, 3, 2))

# Print the second row of the second matrix

print(array\_of\_matrices[2, , 2])

# Print the element in the 3rd row and 3rd column of the 1st matrix

print(array\_of\_matrices[3, 3, 1])

**OUTPUT:**

[1] 13 14 15

[1] 9

**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.**

**CODE:**

# Create three arrays

array1 <- array(1:3, dim = c(1, 3))

array2 <- array(4:6, dim = c(1, 3))

array3 <- array(7:9, dim = c(1, 3))

# Combine the arrays

result <- cbind(array1[1, ], array2[1, ], array3[1, ])

# Print the result

print(result)

**OUTPUT:**

[,1] [,2] [,3]

[1,] 1 4 7

**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.**

**CODE:**

# Create two tables

table1 <- data.frame(col1 = 1:3, col2 = 4:6, col3 = 7:9, col4 = 10:12)

table2 <- data.frame(col1 = 13:15, col2 = 16:18, col3 = 19:21, col4 = 22:24)

# Select the rows and columns to use

rows <- 1:3

cols <- c("col1", "col2", "col3", "col4")

# Convert the tables to arrays

array1 <- as.matrix(table1[rows, cols])

array2 <- as.matrix(table2[rows, cols])

# Combine the arrays into a single array

result <- array(c(array1, array2), dim = c(3, 4, 2))

# Display the content of the array

print(result)

**OUTPUT:**

, , 1

[,1] [,2] [,3] [,4]

[1,] 1 4 7 10

[2,] 2 5 8 11

[3,] 3 6 9 12

, , 2

[,1] [,2] [,3] [,4]

[1,] 13 16 19 22

[2,] 14 17 20 23

[3,] 15 18 21 24

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

**CODE:**

# Create a sequence of even integers greater than 50

even\_ints <- seq(52, 60, by = 2)

# Convert the sequence to a matrix

matrix\_of\_ints <- matrix(even\_ints, nrow = 5, ncol = 3, byrow = TRUE)

# Convert the matrix to an array

array\_of\_ints <- array(matrix\_of\_ints, dim = c(5, 3))

# Display the content of the array

print(array\_of\_ints)

**OUTPUT:**

[,1] [,2] [,3]

[1,] 52 54 56

[2,] 58 60 52

[3,] 54 56 58

[4,] 60 52 54

[5,] 56 58 60

**Use Below Data frame from question 5 to 9**

**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’)**

**)**

**5. Write a R program to extract 3 rd  and 5 th  rows with 1 st  and 3rd  columns from a given data frame**

**CODE:**

# Load the 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")

)

# Extract the 3rd and 5th rows with the 1st and 3rd columns

rows\_to\_extract <- c(3, 5)

cols\_to\_extract <- c(1, 3)

extracted\_data <- exam\_data[rows\_to\_extract, cols\_to\_extract]

# Display the extracted data

print(extracted\_data)

**OUTPUT:**

name attempts

3 Katherine 2

5 Emily 2

**6. 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”)**

**CODE:**

# Load the 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")

)

# Create the new column

country <- c("USA", "USA", "USA", "USA", "UK", "USA", "USA", "India", "USA", "USA")

exam\_data$country <- country

# Display the updated data frame

print(exam\_data)

**OUTPUT:**

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

**7. 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’))**

**CODE:**

# Load the required library

library(dplyr)

# Create the existing 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")

)

# Create the new data frame with the new rows

new\_exam\_data = data.frame(

name = c("Robert", "Sophia"),

score = c(10.5, 9),

attempts = c(1, 3),

qualify = c("yes", "no")

)

# Combine the two data frames using rbind

combined\_exam\_data = rbind(exam\_data, new\_exam\_data)

# Print the result

print(combined\_exam\_data)

**OUTPUT:**

1 Anastasia 12.5 1 yes

2 Dima 9.0 3 no

3 Katherine 16.5 2 yes

4 James 12.0 3 no

5 Emily 9.0 2 no

6 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

**8. Write a R program to sort a given data frame by name and score**

**CODE:**

sorted\_exam\_data = exam\_data %>% arrange(name, score)

print(sorted\_exam\_data)

**OUTPUT:**

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

**9.  Write a R program to save the information of a data frame in a file and display the information of the file.**

**CODE:**

# Load the readr library

library(readr)

# Create the exam data frame

exam\_data = data.frame( name = c("Anastasia", "Dima", "Katherine", "James", "Emily"),

score = c(12.5, 9, 16.5, 12, 9),

attempts = c(1, 3, 2, 3, 2),

qualify = c("yes", "no", "yes", "no", "no") )

# Write the exam data frame to a CSV file

write\_csv(exam\_data, "exam\_data.csv")

# Read the CSV file and display the information

print(read\_csv("exam\_data.csv"))

**10. 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.**

# Load the built-in dataset airquality

airquality = airquality

# Check if it is a data frame

if(is.data.frame(airquality)) {

print("airquality is a data frame")

} else {

print("airquality is not a data frame")

}

# Order the data frame by the first and second column

ordered\_airquality = airquality[order(airquality[,1], airquality[,2]),]

# Remove the variables 'Solar.R' and 'Wind'

airquality\_without\_SolarR\_and\_Wind = ordered\_airquality[

**11. Write a R program to create a factor corresponding to height of women data set , which inbuild in R, contains height and weights for a sample of women.**

**CODE:**

# Load the built-in dataset of heights and weights for women

data("women")

# Create a factor based on the height of women

women$height\_factor = cut(women$height, breaks = c(58, 60, 62, 64), labels = c("Short", "Medium", "Tall"))

# Print the result

print(women)

**12. Write a R program to extract the five of the levels of factor created from a random sample from the LETTERS (Part of the base R distribution.)**

**CODE:**

# Load the required library

library(dplyr)

# Create the factor variable from a random sample of the LETTERS variable

letters\_factor = factor(sample(LETTERS, 50))

# Extract the five levels of the factor variable

levels\_5 = levels(letters\_factor)[1:5]

# Print the result

print(levels\_5)

**13. Iris dataset is a very famous dataset in almost all data mining, machine learning courses, and it has been an R build-in dataset. The dataset consists of 50 samples from each of three species of**

**Iris flowers (Iris setosa, Iris virginica and Iris versicolor). Four features(variables) were**

**measured from each sample, they are the length and the width of sepal and petal, in centimetres.**

**Perform the following EDA steps .**

**# Load the iris dataset**

data("iris")

**# (i) Find dimension, Structure, Summary statistics, Standard Deviation of all features**

dim(iris) # dimensions of the data frame

str(iris) # structure of the data frame

summary(iris) # summary statistics of the data frame

apply(iris[,1:4], 2, sd) # standard deviation of the features

**# (ii) Find mean and standard deviation of features grouped by three species of Iris flowers (Iris setosa, Iris virginica and Iris versicolor)**

group\_by\_species <- group\_by(iris, Species)

summarize(group\_by\_species, mean = mean(Sepal.Length), sd = sd(Sepal.Length))

**# (iii) Find quantile value of sepal width and length**

quantile(iris$Sepal.Width)

quantile(iris$Sepal.Length)

**# (iv) Create new data frame named iris1 which have a new column name Sepal.Length.Cate that categorizes “Sepal.Length” by quantile**

quantile\_cut <- cut(iris$Sepal.Length, quantile(iris$Sepal.Length), labels = c("Q1", "Q2", "Q3", "Q4"))

iris1 <- cbind(iris, Sepal.Length.Cate = quantile\_cut)

**# (v) Average value of numerical variables by two categorical variables: Species and Sepal.Length.Cate**

group\_by\_species\_cate <- group\_by(iris1, Species, Sepal.Length.Cate)

summarize(group\_by\_species\_cate, Sepal.Width\_mean = mean(Sepal.Width), Petal.Length\_mean = mean(Petal.Length), Petal.Width\_mean = mean(Petal.Width))

**# (vi) Average mean value of numerical variables by Species and Sepal.Length.Cate**

group\_by\_species\_cate\_mean <- group\_by(iris1, Species, Sepal.Length.Cate) %>% summarize\_all(mean)

**# (vii) Create Pivot Table based on Species and Sepal.Length.Cate**

pivot\_table <- spread(group\_by\_species\_cate\_mean, key = Sepal.Length.Cate, value = Sepal.Width)

**14.Titanic Casualties – Use the standard ‘Titanic’ dataset which is part of R Base to answer the following questions.**

**# Load Titanic dataset**

data("Titanic")

**# (i) Sum of males vs females aboard**

titanic\_gender\_sum <- with(Titanic, tapply(Freq, Sex, sum))

print(titanic\_gender\_sum)

**# (ii) Sum of survivors vs sex**

titanic\_survived\_sum <- with(Titanic, tapply(Freq, list(Sex, Survived), sum))

print(titanic\_survived\_sum)

**# (iii) Sum of passengers by sex vs age**

titanic\_age\_sum <- with(Titanic, tapply(Freq, list(Sex, Age), sum))

print(titanic\_age\_sum)