20/20

DAY 2 – LAB ASSESSMENT

Reg No: 192121147
Name: Baradh.P

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.

Good

```
vec1 <- c(1,2,3,4,5,6,7,8,9)
vec2 <- c(9,8,9,6,5,4,3,2,1)
matrix1 <- matrix(vec1, nrow=3, ncol=3,
matrix2 <- matrix(vec2, nrow=3, ncol=3
array_mat <- array((matrix1, matrix23, dim=c(3,3,2))
Print carry_mat[2,2]]
Print carry_mat[3,3,1]).
```

Output :-
[1] 4 5 6
[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.

```
arr1<- array(1:9, dim=c(3,3))
arr2 <- array(10:18, dim=((3,3))
arr3 <- array(19:27, dim=((3,3))
Combined_arr <- rbind(arr1[1], arr2[1], arr3[1],
Print Combined_arr
```

Output:

	[1]	[2]	[3]
[1]	1	2	3
[2]	10	11	12
[3]	19	20	21

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.

```
col1 <-c(1,2,3)
col2 <-c(4,5,6)
col3 <-c(7,8,9)
col4 <-c(10,11,12)
row_names <- c("row1", "row2", "row3")
table_names <-c("table1", "table2")
array1 <- array((col1, col2, col3, col4), dim=(3,4,1), dimnames= list
(row_names, (("col1", "col2", "col3", "col4"), table_names[1]))
array2 <- array((col2, col3, col4, col1), dim=(3,4,1), dimnames=
list(row_names, (("col2", "col3", "col4", "col1"), table_names[2]))
Combined_array <- array((array1, array2), dim=((3,4,2), dimnames=
list(row_names, (("col", "col2", "col3", "col4"), table_names)
Print Combined_array)
```

Output:

	[,1]	[,2]	[,3]	[,4]	[,5]
[1]	1	4	7	10	13
[2]	2	5	8	11	14
[3]	3	6	9	12	15.

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

```
arr <- array(seq(from=52, by=2, length.out=15), dim=c(5,3))
print(arr)
```

Output:-

```
     [1] [2] [3]
[1]   52  62  72
[2]   54  64  74
[3]   56  62  78
[4]   58  68  78
[5]   60  70  80.
```

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 3rd and 5th rows with 1st and 3rd columns from a given data frame

```
exam_data[c(3,5),c(1,3)]
```

Output:

3	Katherine	2
5	Emily	2

```
name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'michal', '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, 1.
qual.
```

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")

```
Country <- c("USA", "USA", "USA", "USA", "UK", "USA", "USA", "India",
  "USA", "USA")

exam_data $ Country <- Country.
```

Output:

	Name	Score	attempts	Qualify	Country
1.	Anastasia	12.5	1	yes	USA
2.	Dima	9	3	no	USA
3.	Emily	16.5	2	yes	USA
4.	James	12	3	no	USA
5.	Kevin	8	2	no	USA
6.	Jonas	19.0	1	yes	US

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'))

```
New_data_exam <- data.frame(
name = c("Robert", "Sophia")
Score = (10,5,9)
attempts = c(1,3)
quality = c("yes", "no")
exam_data <- rbind exam_data, new_exam_data)
exam_data
```

Output:

	names	Score	Attempts	quality
1.	Dima	12.5	1	Yes
2.	Katherine	9.0	3	no
3.	James	16.5	2	yes
3			3	yes.
4.	Emily	9.0		

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

```
Sorted_data <- exam_data[order(exam_data Frame $name, exam
- data $ Score),].
```

Output:

	Name	Score	Attempts	quality
1.	Anastasia	12.5	1	yes
2.	Dima	9.0	3	no
3.	James	12.01	2	yes
4.	Emily		2	no
5.	Laura	13.5	1	no
6.	michael		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.

```
write.csv(exam_data, file="exam_data.csv")
exam_data_csv <- read.csv("exam_data.csv")
print(exam_data.csv)
```

Output:

1.	James	12.0	3	no
2.	Laura	13.5	1	no
3.	kavin	8.0	2	no
4.	Jonas	19.0	1	yes.

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.

```
data(airquality)
if(is.data.frame(airquality)){
  print("air quality is a data frame")
}
airquality <- airquality[order(airquality$month, airquality$day),]
airquality <- airquality[,!(names(airquality) %in% c("Solar.R", "Wind"))]
print(airquality).
```

Output:

	Ozone	Solar.R	Wind	Temp	month	Day
	21	1	8	9.7	59	5 9 21
	23	4	25	9.7	61	5 23

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.

```
data(women)
height_factor <- cut(women$height, breaks = seq(50,80,2),
labels=c("50,52", "52-54", "54-56", "56-58", "58-60", "60-62",
"62-64", "64-66", "66-68", "68-70", "70-72", "72-74", "74-76", "76-78",
"78-80")).
levels(height factor).
```

output:

[58,62,7] [62,7,67.3] (67,3,72)
5 5 5

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.)

```
Set.seed(123)
my_letters <- Factor(sample(Letters, 20, replace = TRUE))
my_levels <- levels(my_letters)[1:5]
Print(my levels)
```

output:

[1] "A" "C" "D" "E" "H"

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 .

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

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

(iii)Find quantile value of sepal width and length

(iV)create new data frame named iris1 which have a new column name **Sepal.Length.Cate** that categorizes "Sepal.Length" by quantile

(V) Average value of numerical varialbes by two categorical variables: Species and Sepal.Length.Cate:

(vi) Average mean value of numerical varialbes by Species and Sepal.Length.Cate

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

i) dim (iris)
str (iris)
summary (iris)
apply (iris [, 1:4], 2, sd)

ii) aggregate (iris [, 1:4], by = list (species = iris $ species) Fun = function (x) (mean = mean (x), SD = sd (x)))

iii) quantile (iris $ sepal. width)
quantile (iris $ sepal. length)

iv) aggregate

iris1 <- iris
iris1 $ sepal. length. cate <- cut (iris $ sepal. length, breaks = quantile (iris1 $ sepal. length), lable = FALSE, include. lowest = TRUE)
Iris1 $ sepal. length. (cate <- as. Factor (iris1 $ sepal. length (cate)

v) aggregate (iris1 [1:4], by = list (species = Iris1 $ Species, Sepal. length. (cate = Iris1 $ sepal. length. Cate), Fun = function (x) mean (x))

vi) aggregate (iris1 [, 1:4], by = list [special = iris1 $ Sepeas, sepal. length. (cate), Fun = mean)

Output:

	petal. length	petal. width
1	1.465714	1.0247143
2	3.275000	1.025000
3	4.50000	0.70000
4	5.04000	1.2571429
5	4.050000	2.240000.

vii) library (dplyr)
library (tidyr)
iris1 %>% %>%
group - by (species, Sepal. length. Cate) %>% %>%
Summarize - all (mean) %>% %>%
pivot - longer <- (species, Sepal. length. Cate), names - to = "Variable", value - to = "value").

14.Titanic Casualties – Use the standard 'Titanic' dataset which is part of R Base to answer the following questions.
(i). Use an appropriate apply function to get the sum of males vs females aboard.
(ii). Get a table with the sum of survivors vs sex.
(iii). Get a table with the sum of passengers by sex vs age

i) apply (Titanic, ("Sex", "Survived"), Sum)

ii) addmargins (xtabs) (Freq - Sex + Survived, data: Titanic))

iii) add margins (x tab (Freq - Sex + Age, data = Titanic)).

i) > Grender Count <- apply (Titanic, ((3,4), sum) >> # print the gerder Count > gerder > Count

output:-

Survived	NO	Yes
Child	52	57
Adult	1438	654

ii) # print Survivor by sex <- table (Titanic $ Survived, Titanic $ Sex)

print Survivors the sex Survivors by-sex

	Female	mal
NO	81	468
yes	233	109

iii) # Get Sum passanger Vs age using table Function

Passanger Sex age <- table (Titanic $ Sex, Titanic $ Age Cat)

Output:

	child	Adult
Female	30	152
male	31	398