1 2 3 4

	[,1]	[,2]	[,3]
[1,]	1	3	5
[2,]	2	4	6

function(x)
{
x^2
3

DAY 2 – LAB ASSESSMENT

19/20

Reg No:

Name:

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

Code:

```
mat1 <- matrix(1:9, nrow=3, byrow=TRUE)
mat2 <- matrix(10:18, nrow=3, byrow=TRUE)
```

9

```
arr <- array(c(mat1,mat2),dim=c(3,3,2))
print(arr[2,,2])
print(arr[3,3,1])
```

output

12 84 16

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:

```
arr1<-array(1:9,dim=c(3,3))
arr2<- array(10:18,dim:c(3,3))
arr3<- array(19:27,dim:c(3,3))
combined_arr <- cbind(arr1[1,],arr2[1,], arr3[1,])
print(combined_arr)
```

output:

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

Code:

```
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","Tabel2")
```

```
my_array <- array(data = c(col1, col2, col3, col4), dim = c(3,4,2), dimnames
            =list(row_names, c("col1", "col2", "col3", "col4"), table_names))
print(my_array).
```

output:

Table 1

	col1	col2	col3	col4
Row1	1	4	7	10
Row2	2	5	8	11
Row3	3	6	9	12

Table 2

	col1	col2	col3	col4
Row1	1	4	7	10
Row2	2	5	8	11
Row3	3	6	9	12.

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

code:

```
my_array <- matrix(seq(from=52, by=2, length.out=15), nrow=5)
print(my_array).
```

output:

	[,1]	[,2]	[,3]
[1,]	52	62	72
[2,]	54	64	74
[3,]	56	66	76
[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

extracted -
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:

```
exam_data$country <- c("USA", "USA", "USA", "USA", "UK", "USA", "USA", "India",
                       "USA", "USA")
print (exam_data)
```

output:

	name	score	attempts	qualify	country
a	Anastasia	12.5	1	Yes	USA
d	Dima	9.0	3	no	USA
k	Katherine	16.5	2	Yes	USA
j	James	12.0	3	no	USA
e	Emily	9.0	2	no	UK
m	Michael	20.0	3	Yes	USA
l	Laura	13.5	1	Yes	USA
k	Kevin	8.0	1	no	India
j	Jonas	19.0	2	no	USA
m	Matthew	14.5	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:

```
new_exam_data<- data.frame (name=c('Robert', 'Sophia'),
                            score = c(10.5, 9),
                            attempts = c(1,3),
                            qualify = c('yes', 'no'))
exam_data<-rbind (exam_data, new_exam_data)
print (exam_data).
```

output:

	name	score	attempts	qualify.
	Anastasia	12.5	1	
	Dima	9.0	3	Yes
	Katherine	16.5	2	No
	James	12.0	3	Yes
	Emily	9.0	2	No
	Michael	20.0	3	No
	Matthew	14.0	1	Yes
8	Laura	13.0	1	Yes
9	Kevin	8.0	2	no
10	Jonas	19.0	1	no
11	Robert	10.0	1	yes
				Yes.

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

code:

```
exam_data<-exam_data
  [order(exam_data$name,
         exam_data$score),]

print(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	Katherine	19.0	1	Yes
6	Kevin	16.5	2	yes
7	Jonas	8.0	2	no
8	Laura	13.5	1	no
9	Matthew	14.5	1	yes
10	Michel.	20.5	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:

```
write.csv(exam_data, "exam_data.csv",
          row.names = FALSE)

Exam_data_from_file = read.csv
          ("exam_data.csv")
print(exam_data_from_file)
```

output:

	name	score	attempts	qualify.
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.

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.

code:

```
data(airquality)
if(is.data.frame(airquality))
{
print("airquality is a dataframe.")
}
airquality_ordered <- airquality[order(airquality$Month, airquality$Day),]
airquality_trimmed <- airquality_ordered, c("Ozone", "Month", "Day", "Temp", "Month.1")]
print(airquality_trimmed).
```

output:

"airquality is a data frame"

	Ozone	Month	Day	Temp	Month.1
1	41	5	1	67	5.1
2	36	5	2	72	5.1
3	12	5	3	74	5.1
4	18	5	4	62	5.1
5	NA	5	5	56	5.1
6	28	5	6	66	5.1
7	23	5	7	65	5.1
8	19	5	8	59	5.1
9	8	5	9	61	5.1
10	NA	5	10	69	5.1

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:

```
data(women)
women$height_factor<- cut
        (women$height, breaks=
        quantile(women$height, probs=
                0:4/4)),
labels= c("Short", "Medium-short",
        "Medium-Tall", "Tall"))
head(women)
```

output:

	height	weight	height_factor
1	58	115	Medium-short
2	59	117	Medium-short
3	60	120	Medium-short
4	61	123	Medium-Tall
5	62	126	Medium-Tall
6	63	129	Tall.

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:

```
set.seed(123)
letters_sample<-sample(LETTERS,20)
letters_factor<-factor(letters_sample)
five_levels<-levels(letters_factor)[1:5]
print(five_levels).
```

output:

"A" "B" "C" "D" "E"

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.

code:

```
data(iris)
dim(iris)
str(iris)
summary(iris)
sapply(iris,sd).
aggregate(.~Species,data=iris,mean)
aggregate(iris[,1:4],by=list(Species=iris$Species),sd)
quantile(iris$Sepal.Width)
quantile(iris$Sepal.Length)
iris1<-iris %>%. mutate(Sepal.Length.Cate=cut(Sepal.Length, breaks=quantile
                                              (Sepal.Length)))
aggregate(iris1[,1:4], by=list(Species=iris1$Species, Sepal.Length.Cate=
                               iris1$Sepal.Length.Cate), mean)
aggregate(iris1[,1:4], by=list(species=iris1$Species, Sepal.Length.Cate),
                               FUN=function(x) mean(x))
iris1%.>%. pivot_table(rows=Sepal.Length.Cate, cols=Species, value=
c(Sepal.Width, Sepal.Length, Petal.Width, Petal.~~Width~~length),
                               agg.fun=mean).
```

output:

```
150  5
data(iris)
$Sepal.Length:num  5.1 4.9 4.7 4.6 5 5.4
                   4.6 5 4.4 4.9 ...
$Sepal.Width:num   3.5 3 3.2 3.1 3.6 3.9 3.4
                   3.4 2.9 3.1 ...
$Petal.Length:num  1.4 1.4 1.3 1.5 1.4 1.7
                   1.4 1.5 1.4 1.5 ...
$Petal.Width:num   0.2 0.2 0.2 0.2 0.2
                   0.4 0.3 0.2 0.2 0.1 ...
$Species  : Factor w/ 3 levels ...: 1 1 1 1 1 1 ...
summary(iris)
```

	Sepal.Width	Species
Min. :4.300	Min. :2.000	Setosa :50
1st Qu:5.100	1st Qu:2.800	Versicolor: 50
Median 5.800	Median 3.000	virginica:50
Mean :5.843	Mean :3.057	

1 setosa	5.006	3.428	1.462	0.246
2 versicolour	5.936	2.770	4.260	1.326
3 virginica	6.588	2.974	5.552	2.026

1 Setosa	0.3525	0.3791	0.1737	0.1054
2 Versicolor	0.5162	0.3138	0.4699	0.1978
3 virginica	0.6359	0.3225	0.5519	0.2744

0%	25%	50%	75%	100%
2.000	2.800	3.000	3.300	4.400

0%	25%	50%	75%	100%
4.300	5.100	5.800	6.400	7.900

15 5.8 3.5 1.4 0.2 setosa (4.3, 5.
21 4.9 3.0 1.4 0.2 setosa (4.3, 5.
2 4.7 3.2 1.3 0.2 setosa (4.3, 5.
4 4.6 3.1 1.5 0.2 setosa (4.3, 5.
6 5.0 3.6 1.4 0.2 setosa (4.3, 5.

	Species	Sepal.Length	Petal.
1	setosa	(4.3, 5.1]	0.2
2	versicolor	(4.2, 5.6]	1.00
3	virginica	(4.3, 5.5]	1.3
4	setosa	(5.1, 5.8]	0.2

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

code:

```
data(Titanic)
apply(Titanic, c("Sex", "Survided"), sum)
xtabs(Freq~Survived+Sex, data=Titanic)
xtabs(Freq~Age+Sex, data= margin(Titanic,2))
```

	Survived	
	No	yes
ale	1364	367
male	126	344.

Survived	Sex	
	Male	Female
No	1364	126
yes	367	344.

Sex	
Male	Female
1731	470.

19/20

DAY 3 – LAB ASSESSMENT

Reg No:

N

...ite a function in R programming to print generate Fibonacci sequence using ...sion in R

...de:

```
fib... function (n)
...=0)
...p("Invalid input. n should be a positive integer.")
}
if(n==1)
```