# R PROGRAMMING ASSIGNMENT-5

## 1.Write a program to create a data frame from 4 given vectors?

## ANS:

### R Programming Code:

```
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')

print("Original data frame:")

print(name)

print(score)

print(attempts)

print(qualify)

df = data.frame(name, score, attempts, qualify)

print(df)
```

### OUTPUT:

```
[1] "Original data frame:"

 [1] "Anastasia" "Dima"      "Katherine" "James"     "Emily"     "Michael"

 [7] "Matthew"   "Laura"     "Kevin"     "Jonas"
```

```
 [1] 12.5  9.0 16.5 12.0  9.0 20.0 14.5 13.5  8.0 19.0

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

 [1] "yes" "no"  "yes" "no"  "no"  "yes" "yes" "no"  "no"  "yes"

        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
```

## 2.How are missing values represented in R?

## ANS:

n R, missing values are represented by the symbol **NA** (not available). Impossible values (domain errors like division by 0 et logs of negative numbers are represented by the symbol **NaN** (Not-A-Number). **NA** is used for both numeric and string data.

R being a programming environment, there is no global way to deal with missing values. Many advanced functions, namely modelling functions have several options for dealing with missing values, from pairwise or listwise deletion of missing values to complex multiple imputation methods and have a default method of eliminating missing data. Some functions do have a single missing values option to override the default (produce an **NA** in presence of NAS), many functions have no option at all, i.e. you will have to prepare the data matrix accordingly, selecting only the observations with no missing values.

## 3.Explain what is transpose?

## ANS:

Transpose of a matrix is an operation in which we convert the rows of the matrix in column and column of the matrix in rows

In R, we can transpose data very easily. There are many R packages such as tidyr and reshape2 that helps to reshape data from long to wide format or vice versa. Like many of us, I was also searching transpose function in dplyr package but didn't get any success. Then I realized there is no such function in dplyr to transpose data. Instead there are separate packages built for this function. Both these packages dplyr and tidyr are a part of tidyverse which is a collection of R packages designed to make data manipulation and exploration cakewalk for R users. In this article, i would use these two packages with several examples. These package were written by the most popular R expert **Hadley Wickham**.

## 4.How is data aggregated in R programming?

## ANS:

The process involves two stages. First, collate individual cases of raw data together with a grouping variable. Second, perform which calculation you want on each group of cases. These two stages are wrapped into a single function.

To perform aggregation, we need to specify three things in the code:

- The data that we want to aggregate
- The variable to group by within the data
- The calculation to apply to the groups (what you want to find out)

## The aggregate function:

The first argument to the function is usually a data.frame.

The *by* argument is a list of variables to group by. This must be a list even if there is only one variable, as in the example.

The *FUN* argument is the function which is applied to all columns (i.e., variables) in the grouped data. Because we cannot calculate the average of categorical variables such as *Name* and *Shif*t, they result in empty columns, which I have removed for clarity.

## Other aggregation functions:

Any function that can be applied to a numeric variable can be used within *aggregate*. Maximum, minimum, count, standard deviation and sum are all popular.

For more specific purposes, it is also possible to write your own function in R and refer to that within *aggregate*. I've demonstrated this below where the second largest value of each group is returned, or the largest if the group has only one case. Note also that the groups are formed by *Role* and by *Shift* together.

## 5.What is the subset() function in R?

## ANS:

subset() function in R programming is used to create a subset of vectors, matrices, or data frames based on the conditions provided in the parameters.

## Syntax:

_subset(x, subset, select) Parameters: x: indicates the object. subset: indicates the logical expression on the basis of which subsetting has to be done.

The subset function is available in base R and can be used to return subsets of a vector, martix, or data frame which meet a particular condition. In my three years of using R, I have repeatedly used the subset() function and believe that it is the most useful tool for selecting elements of a data structure. I assume that many of you are familiar with this function, so I will simply conclude this post by providing some brief examples of the subset function.

## 6.Write a program to get the nature of the data of the DataFrame?

## ANS:

We can use class() function to get the nature of the dataframe. It will return: Either data is NULL or not. The datatype of a particular column in a dataframe

## R-PROGRAME:

```
# create vector with names
name = c("sravan","mohan","sudheer",
         "radha","vani","mohan")


# create vector with subjects
subjects = c(".net","Python","java",
             "dbms","os","dbms")


# create a vector with marks
marks=c(98,97,89,90,87,90)


# create vector with height
height=c(5.97,6.11,5.89,
         5.45,5.78,6.0)
```

```
# create vector with weight
weight=c(67,65,78,65,81,76)

# pass these vectors to the data frame
data=data.frame(name,subjects,marks,
                height,weight)

# nature of dataframe
print(paste("names column",class(data$names)))
print(paste("subjects column",class(data$subjects)))
print(paste("marks column",class(data$marks)))
print(paste("height column",class(data$height)))
print(paste("weight column",class(data$weight)))
```

**Output:**

```
[1]  “names column null”
[1]  “subjects column factor”
[1]  “marks column numeric”
[1]  “height column numeric”
[1]  “weight column numeric”
```

## 7.Write a program to get the statistical summary of the data of the Data Frame?

## ANS:

## R Programming Code :

```
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')

)

print("Original dataframe:")

print(exam_data)

print("Statistical summary and nature of the data of the said dataframe:")

print(summary(exam_data))
```

## Output:

```
[1] "Original dataframe:"
       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
[1] "Statistical summary and nature of the data of the said dataframe:"
       name       score          attempts    qualify
 Anastasia:1   Min.   : 8.00   Min.   :1.00   no :5
 Dima     :1   1st Qu.: 9.75   1st Qu.:1.00   yes:5
 Emily    :1   Median :13.00   Median :2.00
 James    :1   Mean   :13.40   Mean   :1.90
 Jonas    :1   3rd Qu.:16.00   3rd Qu.:2.75
 Katherine:1   Max.   :20.00   Max.   :3.00
 (Other)  :4
```