

---

# BIG DATA AND ANALYTICS LAB

---

**(BCSE0183)**

## **Lab Assignment – 03**

**Made by :-** VISHAL DIXIT

**Section :-** B (62)

**University Roll No. :-** 201500792

**Submitted to :-** Dr. Robin Singh Bhadoria

**Functions, Data Frames & R Factor using R**  
**Programming Language**

# 1) Write a R program to perform below operations using R User-Defined Functions

- (a) Creating and Calling user-define Function
- (b) Number of Arguments
- (c) Default Arguments
- (d) Return Values
- (e) Nested Functions
- (f) Recursion

```
Console | Terminal x | Background Jobs x
R 4.2.0 · ~/
> my_function <- function() {
+   print("Hello R Programming!")
+ }
> my_function() # call the function named my_function
[1] "Hello R Programming!"
>
> #function with argument
> my_fun <- function(fname) {
+   paste(fname, "University")
+ }
> my_fun("GLA")
[1] "GLA University"
>
> #Default argument
> my_function <- function(country = "Japan") {
+   paste("I am from", country)
+ }
> my_function("India")
[1] "I am from India"
> my_function() # will get the default value, which is Norway
[1] "I am from Japan"
>
> #Return Value
> my_function <- function(x) {
+   return (5 * x)
+ }
> print(my_function(3))
[1] 15
```

```
> #Nested function
> Outer_func <- function(x) {
+   Inner_func <- function(y) {
+     a <- x + y
+     return(a)
+   }
+   return (Inner_func)
+ }
> output <- Outer_func(3) # To call the Outer_func
> output(5)
[1] 8
>
> #Recursion
> recursion <- function(k) {
+   if (k > 0) {
+     result <- k + recursion(k - 1)
+     print(result)
+   } else {
+     result = 0
+     return(result)
+   }
+ }
> recursion(6)
[1] 1
[1] 3
[1] 6
[1] 10
[1] 15
[1] 21
> |
```

- 2) Write a R program to perform below operations using Data Frames –
- (a) Create Data Frame
  - (b) Summarize the Data
  - (c) Access Items
  - (d) Add Rows & Columns
  - (e) Remove Rows and Columns
  - (f) Amount of Rows and Columns
  - (g) Data Frame Length
  - (h) Combining Data Frames

```
13_09_2022_bigdata.R x  Untitled9* x  Untitled10* x
Source on Save
Run  Source

1 # Create a data frame
2 Data_Frame <- data.frame (
3   Training = c("Strength", "Stamina", "Other"),
4   Pulse = c(100, 150, 120),
5   Duration = c(60, 30, 45)
6 )
7 Data_Frame          # Print the data frame
8
9 summary(Data_Frame)  #summarize the data from a Data Frame
10
11 #Accessing Items
12 Data_Frame[1]
13 Data_Frame[["Training"]]
14 Data_Frame$Training
15
16 #Adding a new row
17 New_row_DF <- rbind(Data_Frame, c("Strength", 110, 110))
18 New_row_DF          # Print the new row
19
20 #Adding a new colm.
21 New_col_DF <- cbind(Data_Frame, Steps = c(1000, 6000, 2000))
22 New_col_DF          # Print the new column
23
24 #Remove rows and colm's
25 Data_Frame_New <- Data_Frame[-c(1), -c(1)]          # Remove the first row and column
26 Data_Frame_New          # Print the new data frame
27
28 #Amount of Rows and Columns
29 dim(Data_Frame)
30
31 #Length of the dataframe
32 length(Data_Frame)
```

```
Console  Terminal x  Background Jobs x
R 4.2.0 · ~/
> Data_Frame          # Print the data frame
  Training Pulse Duration
1 Strength   100     60
2 Stamina   150     30
3 Other     120     45
>
> summary(Data_Frame)  #summarize the data from a Data Frame
  Training      Pulse      Duration
Length:3      Min.   :100.0   Min.   :30.0
Class :character 1st Qu.:110.0   1st Qu.:37.5
Mode  :character Median :120.0   Median :45.0
              Mean  :123.3   Mean  :45.0
              3rd Qu.:135.0   3rd Qu.:52.5
              Max.   :150.0   Max.   :60.0
>
> #Accessing Items
> Data_Frame[1]
  Training
1 Strength
2 Stamina
3 Other
> Data_Frame[["Training"]]
[1] "Strength" "Stamina" "Other"
> Data_Frame$Training
[1] "Strength" "Stamina" "Other"
>
```

### 3) Write a R program to perform below operations using R Factors –

- (a) Create a factor
- (b) Factor Length
- (c) Access Factors
- (d) Change Item Value
- (e) Print the levels using levels( ) function:

```
1 # Create a factor
2 music_genre <- factor(c("Jazz", "Rock", "Classic", "Classic", "Pop", "Jazz", "Rock", "Jazz"))
3
4 music_genre          # Print the factor
5
6 # Factor length
7 length(music_genre)
8
9 #Access factors
10 music_genre[3]
11
12 #Change Item Value
13 music_genre[3] <- "Pop"
14 music_genre[3]
15
16 # levels
17 music_genre <- factor(c("Jazz", "Rock", "Classic", "Classic", "Pop", "Jazz", "Rock", "Jazz"), levels = c("Classic", "Jazz", "Pop", "Rock", "Opera"))
18 music_genre[3] <- "Opera"
19 music_genre[3]
20 levels(music_genre)
```

```
Console Terminal Background Jobs
R 4.2.0 ~ /
> # Create a factor
> music_genre <- factor(c("Jazz", "Rock", "Classic", "Classic", "Pop", "Jazz", "Rock", "Jazz"))
>
> music_genre          # Print the factor
[1] Jazz Rock Classic Classic Pop Jazz Rock Jazz
Levels: Classic Jazz Pop Rock
>
> # Factor length
> length(music_genre)
[1] 8
>
> #Access factors
> music_genre[3]
[1] Classic
Levels: Classic Jazz Pop Rock
>
> #Change Item Value
> music_genre[3] <- "Pop"
> music_genre[3]
[1] Pop
Levels: Classic Jazz Pop Rock
>
> # levels
> music_genre <- factor(c("Jazz", "Rock", "Classic", "Classic", "Pop", "Jazz", "Rock", "Jazz"), levels = c("Classic", "Jazz", "Pop", "Rock", "Opera"))
> music_genre[3] <- "Opera"
> music_genre[3]
[1] Opera
Levels: Classic Jazz Pop Rock Opera
> levels(music_genre)
[1] "Classic" "Jazz" "Pop" "Rock" "Opera"
>
```

#### 4) Create R Factors in Data Frame that prints data with a column of text into categorical form using R Factorial.

```
13_09_2022_bigdata.R x  Untitled9* x  Untitled10* x  Untitled11* x  Untitled12* x
Source on Save  Run
1 # create sample data frame
2 df <- data.frame(x=c(10, 23, 13, 41, 15),
3                 y=c(71, 17, 28, 32, 12))
4
5 # create categorical vector
6 group_vector <- c('A','B','C','D','E')
7
8 # Add categorical variable to the data frame
9 df$group <- factor(group_vector)
10
11 df # print data frame
```

```
Console  Terminal x  Background Jobs x
R 4.2.0 · ~/
>
> df # print data frame
  x  y group
1 10 71    A
2 23 17    B
3 13 28    C
4 41 32    D
5 15 12    E
> |
```