ACADGILD

SESSION 3: FOUNDATIONAL R PROGRAMMING

Assignment 1

PROBLEM STATEMENT :

1. Define an m x n matrix of zeroes and then enters a nested-for loop to fill the locations of the matrix, only if the two indexes differ.

- The purpose is to create a lower triangular matrix, that is a matrix whose elements below the main diagonal are non-zero, the others are left untouched to their initialized zero value.
- When the indexes are equal (if condition in the inner loop, which runs over j, the column index), a break is executed and the innermost loop is interrupted with a direct jump to the instruction following the inner loop, which is a print; then control gets to the outer for condition (over the rows, index i), which is evaluated again.
- If the indexes differ, the assignment is performed and the counter is incremented by 1.
- At the end, the program prints the counter ctr, which contains the #number of elements that were assigned.

SOLUTION :

```
# Acadgild_Data-Analytics_Session-2-Assignment-3.1

# The R-script for the given problem is as follows:

  # Make a lower triangular matrix (zeroes in upper right corner)

  m=8; n=8;

  ctr=0;                  # used to count the assignment

  x_mat = matrix(0,m,n)    # create a 8 x 8 matrix with zeroes

  x_mat

  for(i in 1:m)

   {

     for(j in 1:n)

       {

         if(i==j)             # if the indexes are equal
```

```
        {
          break;
        }
       else                # if the indexes are not equal
        {
          x_mat[i,j] = i+j    #  assign the values only when i<>j
          ctr=ctr+1
        }
     }
    print(i+j)
  }
 print(ctr)              # print how many matrix cells were assigned
 x_mat
```

Here m x n matrix of zeroes is created using matrix(0,m,n); where m=8 and n=8 Hence, 8 X 8 lower triangular matrix is created whose elements below the main diagonal are non-zero, the others are left untouched to their initialized zero value.

When the indexes are equal (i = = j), a break is executed and the innermost loop is interrupted with a direct jump to the instruction following the inner loop, which is a print; then control gets to the outer for condition (over the rows, index i), which is evaluated again.

If the indexes differ (I is not equal to j), the assignment is performed and the counter (ctr) is incremented by 1.

The program prints the counter ctr = 28 (in given sample matrix of order 8 X 8), which contains the number of elements that were assigned.

The final value of x_mat gives the lower triangular matrix.

The Output of R Script in the console are given below:

F:/ACADGILD - Online Course/ACAD Working Directory - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Source

Console Terminal

F:/ACADGILD - Online Course/ACAD Working Directory/

```
> m=8; n=8;
> ctr=0;                    # used to count the assignemnt
> x_mat = matrix(0,m,n)     # create a 8 x 8 matrix with zeroes
> x_mat
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]    0    0    0    0    0    0    0    0
[2,]    0    0    0    0    0    0    0    0
[3,]    0    0    0    0    0    0    0    0
[4,]    0    0    0    0    0    0    0    0
[5,]    0    0    0    0    0    0    0    0
[6,]    0    0    0    0    0    0    0    0
[7,]    0    0    0    0    0    0    0    0
[8,]    0    0    0    0    0    0    0    0
> for(i in 1:m)
+ {
+
+     for(j in 1:n)
+
+     {
+
+       if(i==j)              # if the indexes are equal
+
+       {
+
+         break;
+
+       }
+
```

F:/ACADGILD - Online Course/ACAD Working Directory - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Source

Console Terminal

F:/ACADGILD - Online Course/ACAD Working Directory/

```
+
+       else                  # if the indexes are not equal
+
+       {
+
+         x_mat[i,j] = i+j    #  assign the values only when i<>j
+
+         ctr=ctr+1
+
+       }
+
+     }
+
+     print(i+j)
+
+   }
[1] 2
[1] 4
[1] 6
[1] 8
[1] 10
[1] 12
[1] 14
[1] 16
> print(ctr)                # print how many matrix cells were assigned
[1] 28
> x_mat
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]    0    0    0    0    0    0    0    0
[2,]    3    0    0    0    0    0    0    0
[3,]    4    5    0    0    0    0    0    0
[4,]    5    6    7    0    0    0    0    0
[5,]    6    7    8    9    0    0    0    0
[6,]    7    8    9   10   11    0    0    0
[7,]    8    9   10   11   12   13    0    0
[8,]    9   10   11   12   13   14   15    0
>
```