



**ACADGILD**

## SESSION 3: Assignment 1

---

Solution — Guru Pra

## Table of Contents

1.Problem Statement .....	3
2.Solution .....	3

## 1. Problem Statement

1. Define an  $m \times n$  matrix of zeros 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.

```

for (i in 1:m) {
  if(i==j)
    break
  }else
  {
    x_mat[i,j] = i+j
    ctr=ctr+1
  }
}
print(i+j)
}
print(ctr)
x_mat
}

```

## 2. Solution

```

2. #Vectorized form
set.seed(42)

#create matrix
mat_1<- replicate(10,rnorm(10))

#transform into data frame
df_1= data.frame(mat_1)
df_1<- df_1 + 10*sin(0.75*pi)

#non-vectorized form
set.seed(42)
#create matrix
mat_1<- replicate(10,rnorm(10))
#transform into data frame
df_1= data.frame(mat_1)

for(i in 1:10){
  for(j in 1:10){
    df_1[i,j]<- df_1[i,j] + 10*sin(0.75*pi)
    print(df_1)
  }
}

#time difference

system.time(
  df_1[i,j]<- df_1[i,j] + 10*sin(0.75*pi)
)

system.time(
  for(i in 1:10){
    for(j in 1:10){
      df_1[i,j]<- df_1[i,j] + 10*sin(0.75*pi)
    }
  }
)

```

```

3. mymat <-matrix(rep(1:5,4),ncol=4)
mymatsum_row_mymat <- apply(mymat,1,sum)
sum_col_mymat <- apply(mymat,2,sum)
a <- cbind(mymatsum_row_mymat,

```