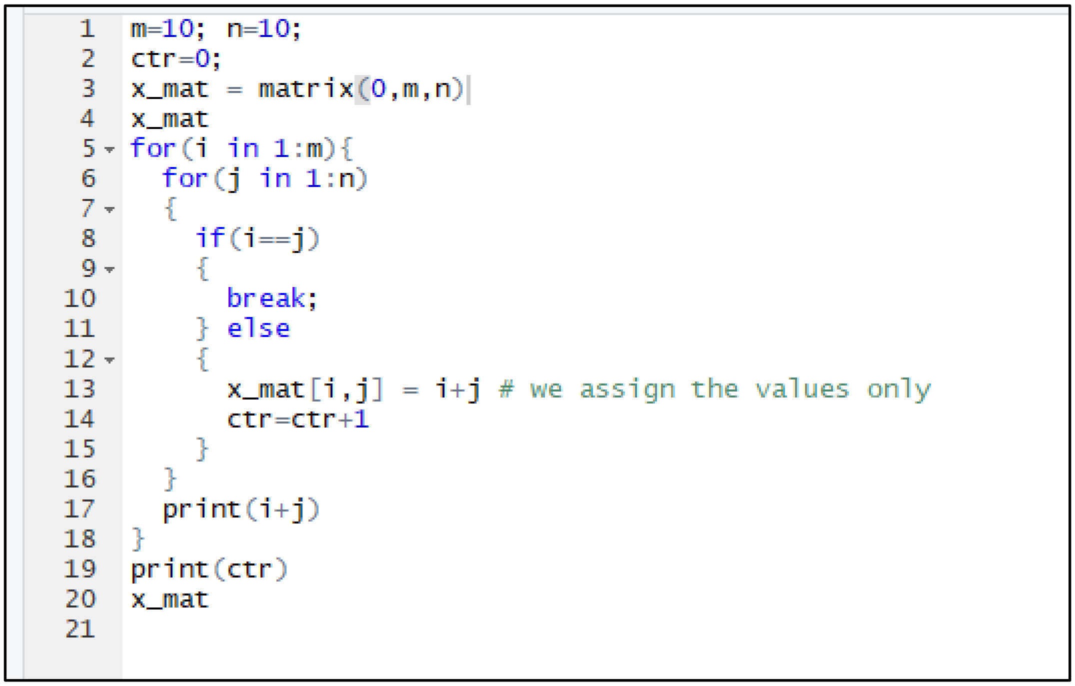
SESSION 3 – ASSIGNMENT 3.1

Date: 29th December 2018

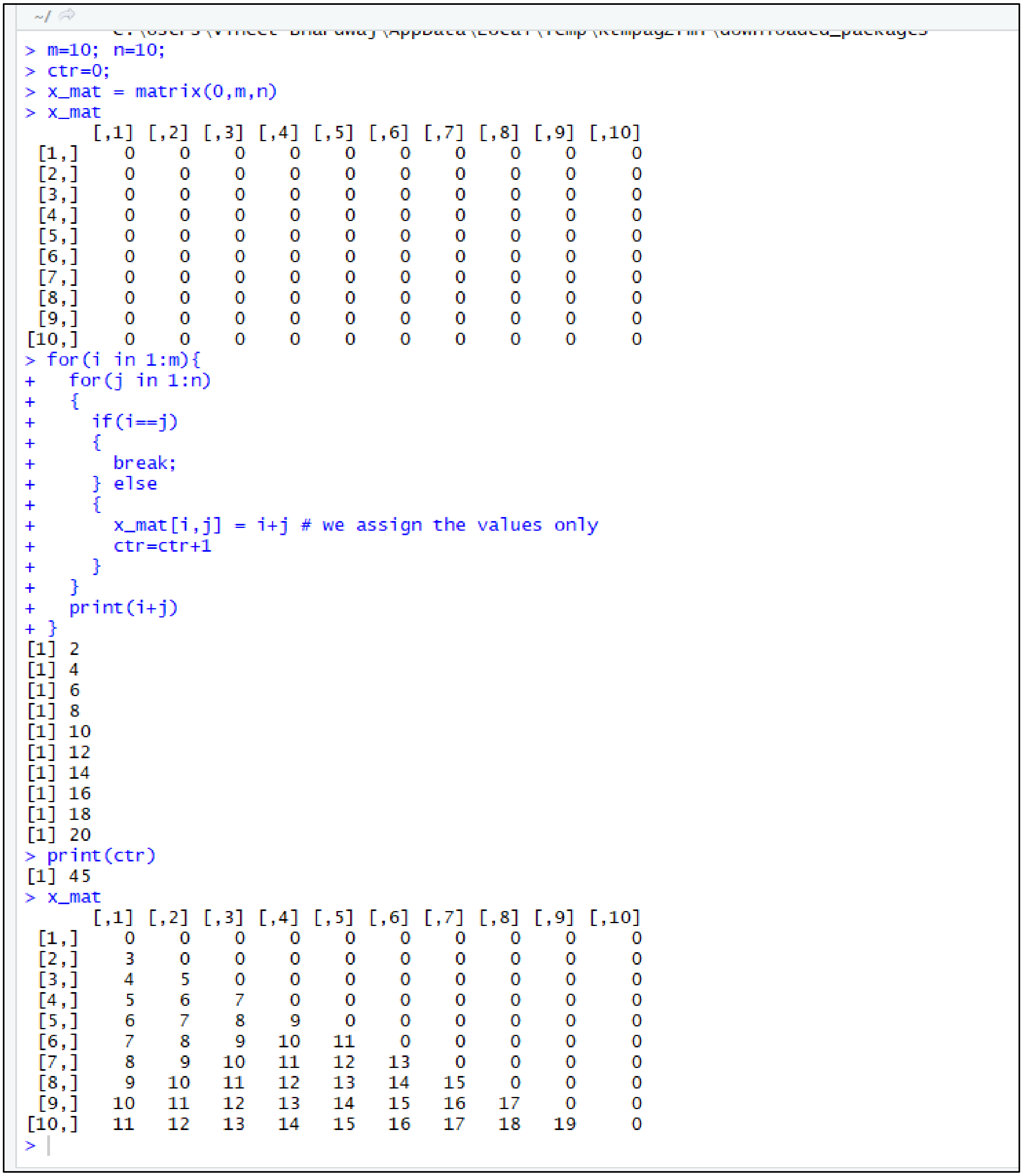
1. **Define an m x n matrix of zeros and then enters a nested-for loop to fill the locations of the matrix, only if the two indexes differ.**

- 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.



```
m=10; n=10;
ctr=0;
x_mat = matrix(0,m,n)
x_mat
for(i in 1:m){
  for(j in 1:n)
  {
    if(i==j)
    {
      break;
    } else
    {
      x_mat[i,j] = i+j # we assign the values only
      ctr=ctr+1
    }
  }
  print(i+j)
}
print(ctr)
x_mat

```



```
      C:\Users\Vineet Bhardwaj\AppData\Local\Temp\RtmpagZ1mn\downloaded_packages
> m=10; n=10;
> ctr=0;
> x_mat = matrix(0,m,n)
> x_mat
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    0    0    0    0    0    0    0    0    0     0
 [2,]    0    0    0    0    0    0    0    0    0     0
 [3,]    0    0    0    0    0    0    0    0    0     0
 [4,]    0    0    0    0    0    0    0    0    0     0
 [5,]    0    0    0    0    0    0    0    0    0     0
 [6,]    0    0    0    0    0    0    0    0    0     0
 [7,]    0    0    0    0    0    0    0    0    0     0
 [8,]    0    0    0    0    0    0    0    0    0     0
 [9,]    0    0    0    0    0    0    0    0    0     0
[10,]    0    0    0    0    0    0    0    0    0     0
> for(i in 1:m){
+   for(j in 1:n)
+   {
+     if(i==j)
+     {
+       break;
+     } else
+     {
+       x_mat[i,j] = i+j # we assign the values only
+       ctr=ctr+1
+     }
+   }
+   print(i+j)
+ }
[1] 2
[1] 4
[1] 6
[1] 8
[1] 10
[1] 12
[1] 14
[1] 16
[1] 18
[1] 20
> print(ctr)
[1] 45
> x_mat
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    0    0    0    0    0    0    0    0    0     0
 [2,]    3    0    0    0    0    0    0    0    0     0
 [3,]    4    5    0    0    0    0    0    0    0     0
 [4,]    5    6    7    0    0    0    0    0    0     0
 [5,]    6    7    8    9    0    0    0    0    0     0
 [6,]    7    8    9   10   11    0    0    0    0     0
 [7,]    8    9   10   11   12   13    0    0    0     0
 [8,]    9   10   11   12   13   14   15    0    0     0
 [9,]   10   11   12   13   14   15   16   17    0     0
[10,]   11   12   13   14   15   16   17   18   19     0
> 
```