

# SESSION 3: FOUNDATIONALR PROGRAMMING

## Assignment 2

1. Create an  $m \times n$  matrix with `replicate(m, rnorm(n))` with  $m=10$  column vectors of  $n=10$  elements each, constructed with `rnorm(n)`, which creates random normal numbers.
  - Then we transform it into a dataframe (thus 10 observations of 10 variables) and perform an algebraic operation on each element using a nested for loop: at each iteration, every element referred by the two indexes is incremented by a sinusoidal function, compare the vectorized and non-vectorized form of creating the solution and report the system time differences.

Answer:

```
m <- 10
n <- 10
sinVal <- sin(.25*pi)
```

```
df<- as.data.frame(matrix (replicate(m, rnorm(n)) , m, n))
df
```

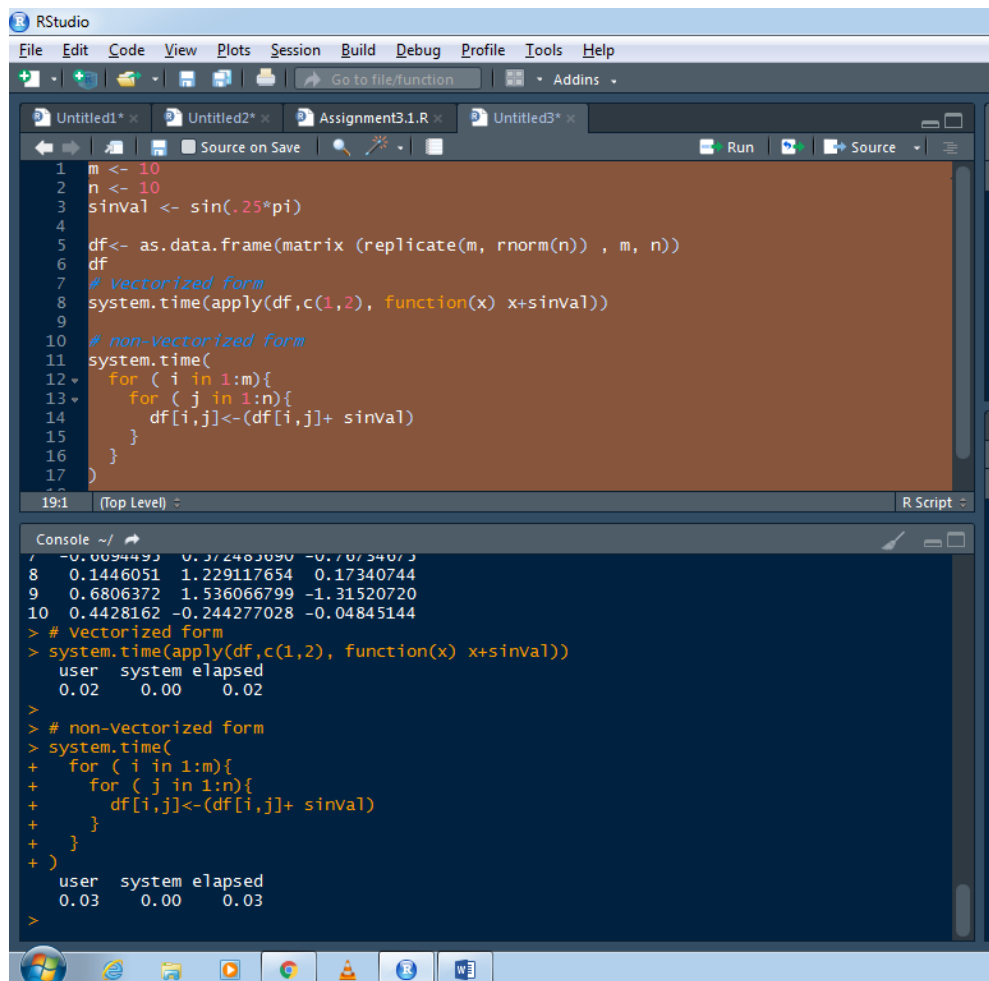
**# Vectorized form**

```
system.time(apply( df , c(1,2), function(x) x + sinVal ) )
```

**# non-Vectorized form**

```
system.time(
  for ( i in 1:m){
    for ( j in 1:n){
      df[i,j]<-(df[i,j]+ sinVal)
    }
  }
)
```

Output:



The screenshot shows the RStudio interface. The script editor contains the following code:

```
1 m <- 10
2 n <- 10
3 sinVal <- sin(.25*pi)
4
5 df<- as.data.frame(matrix (replicate(m, rnorm(n)) , m, n))
6 df
7 # Vectorized form
8 system.time(apply(df,c(1,2), function(x) x+sinval))
9
10 # non-vectorized form
11 system.time(
12   for ( i in 1:m){
13     for ( j in 1:n){
14       df[i,j]<-(df[i,j]+ sinval)
15     }
16   }
17 )
```

The console output shows the initial matrix and the execution times for both methods:

```
19:1 (Top Level) >
7 -0.0094493  0.372483090 -0.70734073
8  0.1446051  1.229117654  0.17340744
9  0.6806372  1.536066799 -1.31520720
10 0.4428162 -0.244277028 -0.04845144
> # Vectorized form
> system.time(apply(df,c(1,2), function(x) x+sinval))
   user system elapsed 
0.02   0.00   0.02 
> 
> # non-vectorized form
> system.time(
+   for ( i in 1:m){
+     for ( j in 1:n){
+       df[i,j]<-(df[i,j]+ sinval)
+     }
+   }
+ )
   user system elapsed 
0.03   0.00   0.03 
>
```