Problem Statement

1. Create an m x 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: Here, we need to compare the system time differences and the solutions for a vector m x n with 10 variables which is converted to data frame between for loop and vectorization form.

Vectorization

Step 1: Creating matrix using replicate function. Before using rnorm to generate random number , use set.seed() function.

set.seed(42)

mat1<- replicate(10,rnorm(10))

Step 2: Transform into data frame

df1<- data.frame(mat1)

Step 3: Apply the sinusoidal function on the variable of the df1

df1<- df1+ 10*sin(0.75*pi)

For Loop:

Step 1: Creating matrix using replicate function.

mat1<- replicate(10,rnorm(10))

Step 2: Transform into data frame

df1<- data.frame(mat1)

Step3: For Loop

```
for(i in 1:10){
 for(j in 1:10){
  df1[i,j]<- df1[i,j]+ 10*sin(0.75*pi)
  print(df1)
 }
}
```

Step 4: Time difference using system.time

For Vectorization:

system.time(df1+ 10*sin(0.75*pi))

Output obtained in quick time.

```
> #time difference
> system.time(df1+ 10*sin(0.75*pi))
   user  system elapsed
      0       0       0
> |
```

For loop:

```
        system.time(for(i in 1:10){
  for(j in 1:10){
   df1[i,j]<- df1[i,j]+ 10*sin(0.75*pi)
   print(df1)
  }
})
```

Time taken is much more compared to vectorized data frame.

```
 user   system elapsed
 1.27    0.00    1.32
```

R Script is uploaded for reference.