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.

Ans:

> set.seed(42);

> m=10; n=10;

> mymat<-replicate(m, rnorm(n)) # create matrix of normal random numbers

> mydframe=data.frame(mymat) # transform into data frame

> #we can use system.stem() to check how long this takes

> system.time(for (i in 1:m) {

+ for (j in 1:n) {

+ mydframe[i,j]<-mydframe[i,j] + 10\*sin(0.75\*pi)

+ }

+ }

+ )

user system elapsed

0.03 0.00 0.03

> mydframe

X1 X2 X3 X4 X5 X6 X7 X8 X9

1 8.442026 8.375937 6.764429 7.526518 7.277066 7.392993 6.703833 6.027949 8.583775

2 6.506370 9.357713 5.289759 7.775905 6.710011 6.287229 7.256298 6.980881 7.328989

3 7.434196 5.682207 6.899150 8.106171 7.829231 8.646795 7.652892 7.694586 7.159508

4 7.703930 6.792279 8.285743 6.462141 6.344363 7.713967 8.470805 6.117544 6.950171

5 7.475336 6.937746 8.966261 7.576023 5.702787 7.160828 6.343776 6.528239 5.876739

6 6.964943 7.707018 6.640599 5.354059 7.503886 7.347619 8.373610 7.652064 7.683065

7 8.582590 6.786815 6.813798 6.286609 6.259675 7.750357 7.406916 7.839247 6.853928

8 6.976409 4.414612 5.307905 6.220160 8.515169 7.160901 8.109574 7.534835 6.888311

9 9.089492 4.630601 7.531165 4.656860 6.639622 4.077978 7.991796 6.185292 8.004414

10 7.008354 8.391181 6.431073 7.107190 7.726716 7.355951 7.791946 5.971287 7.892841

X10

1 8.463184

2 6.594894

3 7.721416

4 8.462178

5 5.960279

6 6.210275

7 5.939329

8 5.611854

9 7.151050

10 7.724272

> #### vector version

> set.seed(42);

> m=10; n=10;

> mymat<-replicate(m, rnorm(n))

> mydframe=data.frame(mymat)

> system.time(mydframe<-mydframe + 10\*sin(0.75\*pi))

user system elapsed

0 0 0

> mydframe

X1 X2 X3 X4 X5 X6 X7 X8 X9

1 8.442026 8.375937 6.764429 7.526518 7.277066 7.392993 6.703833 6.027949 8.583775

2 6.506370 9.357713 5.289759 7.775905 6.710011 6.287229 7.256298 6.980881 7.328989

3 7.434196 5.682207 6.899150 8.106171 7.829231 8.646795 7.652892 7.694586 7.159508

4 7.703930 6.792279 8.285743 6.462141 6.344363 7.713967 8.470805 6.117544 6.950171

5 7.475336 6.937746 8.966261 7.576023 5.702787 7.160828 6.343776 6.528239 5.876739

6 6.964943 7.707018 6.640599 5.354059 7.503886 7.347619 8.373610 7.652064 7.683065

7 8.582590 6.786815 6.813798 6.286609 6.259675 7.750357 7.406916 7.839247 6.853928

8 6.976409 4.414612 5.307905 6.220160 8.515169 7.160901 8.109574 7.534835 6.888311

9 9.089492 4.630601 7.531165 4.656860 6.639622 4.077978 7.991796 6.185292 8.004414

10 7.008354 8.391181 6.431073 7.107190 7.726716 7.355951 7.791946 5.971287 7.892841

X10

1 8.463184

2 6.594894

3 7.721416

4 8.462178

5 5.960279

6 6.210275

7 5.939329

8 5.611854

9 7.151050

10 7.724272