# 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.

m<-replicate(10,rnorm(10),simplify="matrix")

View(m)

df<-as.data.frame(m)

View(df)

vect = as.vector(m)

View(vect)

start.time.normal<-Sys.time() #gives time at the starting of the process

for(i in seq(nrow(m))){

for(j in seq(ncol(m))){

print(2\*sin(m[i,j]))

}}

end.time.normal<-Sys.time() #gives time at the end of the process

end.time.normal-start.time.normal # diff gives time taken during the process. for this, the function to be selected all and then run

|  |
| --- |
| > m<-replicate(10,rnorm(10),simplify="matrix")  > View(m)  > df<-as.data.frame(m)  > View(df)  > vect = as.vector(m)  > View(vect)  >  > start.time.normal<-Sys.time() #gives time at the starting of the process  > for(i in seq(nrow(m))){  + for(j in seq(ncol(m))){  + print(2\*sin(m[i,j]))  + }}  [1] -1.807292  [1] -1.030394  [1] -1.964089  [1] 0.6745816  [1] 1.1637  [1] -0.02708439  [1] 0.4854453  [1] 1.769076  [1] -1.985387  [1] -0.4568946  [1] -0.7743087  [1] 0.08293838  [1] 1.758052  [1] 1.621647  [1] 0.5605196  [1] -1.642664  [1] -0.3149207  [1] 1.950328  [1] 1.239418  [1] 1.178032  [1] 1.756836  [1] -0.863908  [1] -1.660958  [1] -1.668011  [1] 1.990852  [1] 0.8597298  [1] 1.822193  [1] -1.436256  [1] -1.99698  [1] 1.741227  [1] -1.25386  [1] -0.03455011  [1] -1.041316  [1] 0.5557742  [1] -0.672947  [1] 1.225158  [1] 1.229278  [1] 1.455644  [1] 1.874858  [1] 0.3196811  [1] -1.752956  [1] 0.41574  [1] 1.049511  [1] 0.2759229  [1] -1.313768  [1] -1.61562  [1] -1.757401  [1] 1.883453  [1] 1.841691  [1] 1.808582  [1] 1.112138  [1] -1.162182  [1] 1.94927  [1] -1.741853  [1] -0.4049106  [1] -1.997171  [1] 0.7380568  [1] 1.924078  [1] 1.948307  [1] -0.571025  [1] 1.997702  [1] 1.535683  [1] 0.4974029  [1] -1.92288  [1] 0.2197683  [1] 0.4552401  [1] -0.5567943  [1] -0.5792825  [1] -0.2107935  [1] 1.310054  [1] -1.924211  [1] 1.53293  [1] 1.987089  [1] -0.5016231  [1] -0.158386  [1] -0.3328866  [1] -1.980042  [1] 1.36465  [1] -0.7523421  [1] -0.8294757  [1] 1.082989  [1] 0.5524222  [1] -1.890412  [1] 1.305341  [1] 0.5756954  [1] -1.398854  [1] -1.938406  [1] -0.2962121  [1] -1.316969  [1] 1.971857  [1] -1.883253  [1] 0.3137295  [1] 1.649785  [1] 1.655185  [1] -0.2988315  [1] -0.5447062  [1] 0.2763466  [1] -1.999935  [1] -0.05125753  [1] -1.746422  > end.time.normal<-Sys.time() #gives time at the end of the process  >  > end.time.normal-start.time.normal # diff gives time taken during the process.  Time difference of 0.06617594 secs |
|  |
| |  | | --- | | > | |

start.time.vect<-Sys.time()#gives time at the starting of the process

print(2\*sin(vect))

end.time.vect<-Sys.time()#gives time at the end of the process

end.time.vect-start.time.vect # diff gives time taken during the process. for this, the function to be selected all and then run system time differences.

end.time.normal-start.time.normal

|  |
| --- |
| > start.time.vect<-Sys.time()#gives time at the starting of the process  > print(2\*sin(vect))  [1] -1.80729213 -0.77430869 1.75683560 -1.25385979 -1.75295606 1.11213837  [7] 1.99770215 -1.92421086 1.08298922 -1.88325281 -1.03039360 0.08293838  [13] -0.86390800 -0.03455011 0.41573999 -1.16218192 1.53568268 1.53292995  [19] 0.55242220 0.31372955 -1.96408857 1.75805154 -1.66095779 -1.04131646  [25] 1.04951073 1.94926969 0.49740290 1.98708894 -1.89041153 1.64978475  [31] 0.67458161 1.62164694 -1.66801130 0.55577424 0.27592287 -1.74185308  [37] -1.92288027 -0.50162309 1.30534078 1.65518516 1.16369956 0.56051960  [43] 1.99085227 -0.67294697 -1.31376822 -0.40491056 0.21976835 -0.15838597  [49] 0.57569535 -0.29883148 -0.02708439 -1.64266366 0.85972983 1.22515806  [55] -1.61562047 -1.99717086 0.45524008 -0.33288663 -1.39885364 -0.54470616  [61] 0.48544532 -0.31492070 1.82219273 1.22927834 -1.75740096 0.73805684  [67] -0.55679430 -1.98004156 -1.93840609 0.27634660 1.76907612 1.95032765  [73] -1.43625608 1.45564353 1.88345258 1.92407811 -0.57928247 1.36465009  [79] -0.29621212 -1.99993452 -1.98538672 1.23941755 -1.99697957 1.87485786  [85] 1.84169091 1.94830661 -0.21079353 -0.75234208 -1.31696881 -0.05125753  [91] -0.45689462 1.17803214 1.74122725 0.31968110 1.80858232 -0.57102498  [97] 1.31005449 -0.82947572 1.97185662 -1.74642166  > end.time.vect<-Sys.time()#gives time at the end of the process  > end.time.vect-start.time.vect # diff gives time taken during the process.  Time difference of 0.02807093 secs  > #for this, the function to be selected all and then run system time differences.  > end.time.normal-start.time.normal  Time difference of 0.06617594 secs |
|  |
| |  | | --- | | > | |

## same problem solved by using benchmark function, easier and gives result in data frame

install.packages("rbenchmark")

library(rbenchmark)

View(benchmark (

NonVector=for(i in seq(nrow(m))){

for(j in seq(ncol(m))){

print(2\*sin(m[i,j]))

}},

forvector=print(2\*sin(vect)))

)

|  |
| --- |
| 2 |