**ACD\_BART1\_Session\_2\_Assignment\_3\_Main**

**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.**

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

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

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

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)))

)