Assignment3.2

Aruna

24 November 2018

```
# 2. 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.


set.seed(2)
#create matrix
matrix_1<- replicate(10,rnorm(10))
matrix_1
```

```
##              [,1]        [,2]         [,3]       [,4]        [,5]
##  [1,] -0.89691455  0.41765075  2.090819205  0.7389386 -0.38358623
##  [2,]  0.18484918  0.98175278 -1.199925820  0.3189604 -1.95910318
##  [3,]  1.58784533 -0.39269536  1.589638200  1.0761644 -0.84170506
##  [4,] -1.13037567 -1.03966898  1.954651642 -0.2841577  1.90354747
##  [5,] -0.08025176  1.78222896  0.004937777 -0.7766753  0.62249393
##  [6,]  0.13242028 -2.31106908 -2.451706388 -0.5956605  1.99092044
##  [7,]  0.70795473  0.87860458  0.477237303 -1.7259798 -0.30548372
##  [8,] -0.23969802  0.03580672 -0.596558169 -0.9025845 -0.09084424
##  [9,]  1.98447394  1.01282869  0.792203270 -0.5590619 -0.18416145
## [10,] -0.13878701  0.43226515  0.289636710 -0.2465126 -1.19876777
##             [,6]       [,7]        [,8]       [,9]        [,10]
##  [1,] -0.8382871 -1.7882422 -0.92127567  0.9959846  1.600390852
##  [2,]  2.0663014  2.0312425  0.33044950 -1.6957649  1.681154956
##  [3,] -0.5622471 -0.7031443 -0.14166081 -0.5333721 -1.183606388
##  [4,]  1.2757155  0.1581648  0.43484776 -1.3722695 -1.358457254
##  [5,] -1.0475726  0.5062348 -0.05372263 -2.2079198 -1.512670795
##  [6,] -1.9658782 -0.8199951 -0.90711038  1.8221225 -1.253104899
##  [7,] -0.3229711 -1.9988470  1.30351223 -0.6533934  1.959357077
##  [8,]  0.9358625 -0.4792926  0.77178978 -0.2846812  0.007645872
##  [9,]  1.1392298  0.0841799  1.05252560 -0.3869496 -0.842615198
## [10,]  1.6716188 -0.8954866 -1.41003834  0.3866950 -0.601160105
```

```
#transform into data frame
df_1= data.frame(matrix_1)
```

```r
# using Sin function
df_1<- df_1 + 20*sin(pi/2)
df_1
```

```
##          X1       X2       X3       X4       X5       X6       X7       X8
## 1  19.10309 20.41765 22.09082 20.73894 19.61641 19.16171 18.21176 19.07872
## 2  20.18485 20.98175 18.80007 20.31896 18.04090 22.06630 22.03124 20.33045
## 3  21.58785 19.60730 21.58964 21.07616 19.15829 19.43775 19.29686 19.85834
## 4  18.86962 18.96033 21.95465 19.71584 21.90355 21.27572 20.15816 20.43485
## 5  19.91975 21.78223 20.00494 19.22332 20.62249 18.95243 20.50623 19.94628
## 6  20.13242 17.68893 17.54829 19.40434 21.99092 18.03412 19.18000 19.09289
## 7  20.70795 20.87860 20.47724 18.27402 19.69452 19.67703 18.00115 21.30351
## 8  19.76030 20.03581 19.40344 19.09742 19.90916 20.93586 19.52071 20.77179
## 9  21.98447 21.01283 20.79220 19.44094 19.81584 21.13923 20.08418 21.05253
## 10 19.86121 20.43227 20.28964 19.75349 18.80123 21.67162 19.10451 18.58996
##          X9      X10
## 1  20.99598 21.60039
## 2  18.30424 21.68115
## 3  19.46663 18.81639
## 4  18.62773 18.64154
## 5  17.79208 18.48733
## 6  21.82212 18.74690
## 7  19.34661 21.95936
## 8  19.71532 20.00765
## 9  19.61305 19.15738
## 10 20.38669 19.39884
```

```r
#non-vectorized form
set.seed(2)
#create matrix
matrix_2<- replicate(10,rnorm(10))
#transform into data frame
df_2= data.frame(matrix_2)
for(i in 1:10){
  for(j in 1:10){
    df_2[i,j]<- df_1[i,j] + 20*sin(pi/2)
   # print(df_2)
  }
}
print(df_2)
```

```
##          X1       X2       X3       X4       X5       X6       X7       X8
## 1  39.10309 40.41765 42.09082 40.73894 39.61641 39.16171 38.21176 39.07872
## 2  40.18485 40.98175 38.80007 40.31896 38.04090 42.06630 42.03124 40.33045
## 3  41.58785 39.60730 41.58964 41.07616 39.15829 39.43775 39.29686 39.85834
## 4  38.86962 38.96033 41.95465 39.71584 41.90355 41.27572 40.15816 40.43485
## 5  39.91975 41.78223 40.00494 39.22332 40.62249 38.95243 40.50623 39.94628
## 6  40.13242 37.68893 37.54829 39.40434 41.99092 38.03412 39.18000 39.09289
## 7  40.70795 40.87860 40.47724 38.27402 39.69452 39.67703 38.00115 41.30351
## 8  39.76030 40.03581 39.40344 39.09742 39.90916 40.93586 39.52071 40.77179
```

```
## 9  41.98447 41.01283 40.79220 39.44094 39.81584 41.13923 40.08418 41.05253
## 10 39.86121 40.43227 40.28964 39.75349 38.80123 41.67162 39.10451 38.58996
##          X9      X10
## 1  40.99598 41.60039
## 2  38.30424 41.68115
## 3  39.46663 38.81639
## 4  38.62773 38.64154
## 5  37.79208 38.48733
## 6  41.82212 38.74690
## 7  39.34661 41.95936
## 8  39.71532 40.00765
## 9  39.61305 39.15738
## 10 40.38669 39.39884

#time difference
system.time(df_1[i,j]<- df_1[i,j] + 20*sin(pi/2))

##    user  system elapsed
##       0       0       0

system.time(for(i in 1:10)
              {
                for(j in 1:10)
                  {
                    df_2[i,j]<- df_2[i,j] + 20*sin(pi/2)
                   }
              }
            )

##    user  system elapsed
##       0       0       0
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.