

Homework_1

LekhShetty

Question 2

Part 1

```
my_vec <- c(
  "+0.07",
  "-0.07",
  "+0.25",
  "-0.84",
  "+0.32",
  "-0.24",
  "-0.97",
  "-0.36",
  "+1.76",
  "-0.36"
)
typeof(my_vec)
```

```
[1] "character"
```

```
"The data types of the elements in the vector are characters"
```

```
[1] "The data types of the elements in the vector are characters"
```

Part 2

```
my_vec_double <- as.double(my_vec)
my_vec_double
```

```
[1] 0.07 -0.07 0.25 -0.84 0.32 -0.24 -0.97 -0.36 1.76 -0.36
```

```
my_vec_integer <- as.integer(my_vec)
my_vec_integer
```

```
[1] 0 0 0 0 0 0 0 0 1 0
```

Part 3

```
my_vec_bool <- sum((ifelse(my_vec_double <= 0, TRUE, FALSE))==FALSE)
my_vec_bool
```

```
[1] 4
```

Part 4

```
my_vec_double <- sort(my_vec_double)
my_vec_double
```

```
[1] -0.97 -0.84 -0.36 -0.36 -0.24 -0.07 0.07 0.25 0.32 1.76
```

Question 3

Part 1

```
matrix_1 = matrix(c(1, 2, 3, 4, 5, 6, 7, 8, 9), nrow=3, ncol=3, byrow=TRUE)
matrix_1
```

```

      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9

```

```

row1 <- matrix(c(1:100), nrow=1, ncol=100)
row2 <- matrix(c((1:100)**2), nrow=1, ncol=100)
matrix_2 <- rbind(row1, row2)
matrix_2

```

```

      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
[1,]    1    2    3    4    5    6    7    8    9   10   11   12   13   14
[2,]    1    4    9   16   25   36   49   64   81   100  121  144  169  196
      [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25] [,26]
[1,]    15   16   17   18   19   20   21   22   23   24   25   26
[2,]   225  256  289  324  361  400  441  484  529  576  625  676
      [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37] [,38]
[1,]    27   28   29   30   31   32   33   34   35   36   37   38
[2,]   729  784  841  900  961 1024 1089 1156 1225 1296 1369 1444
      [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49] [,50]
[1,]    39   40   41   42   43   44   45   46   47   48   49   50
[2,]  1521 1600 1681 1764 1849 1936 2025 2116 2209 2304 2401 2500
      [,51] [,52] [,53] [,54] [,55] [,56] [,57] [,58] [,59] [,60] [,61] [,62]
[1,]    51   52   53   54   55   56   57   58   59   60   61   62
[2,]  2601 2704 2809 2916 3025 3136 3249 3364 3481 3600 3721 3844
      [,63] [,64] [,65] [,66] [,67] [,68] [,69] [,70] [,71] [,72] [,73] [,74]
[1,]    63   64   65   66   67   68   69   70   71   72   73   74
[2,]  3969 4096 4225 4356 4489 4624 4761 4900 5041 5184 5329 5476
      [,75] [,76] [,77] [,78] [,79] [,80] [,81] [,82] [,83] [,84] [,85] [,86]
[1,]    75   76   77   78   79   80   81   82   83   84   85   86
[2,]  5625 5776 5929 6084 6241 6400 6561 6724 6889 7056 7225 7396
      [,87] [,88] [,89] [,90] [,91] [,92] [,93] [,94] [,95] [,96] [,97] [,98]
[1,]    87   88   89   90   91   92   93   94   95   96   97   98
[2,]  7569 7744 7921 8100 8281 8464 8649 8836 9025 9216 9409 9604
      [,99] [,100]
[1,]    99   100
[2,]  9801 10000

```

Part 2

```
row_wise_scan <- function(x){
  n <- nrow(x)
  m <- ncol(x)

  # Insert your code here
  count <- 0
  for(i in 1:n){
    for(j in 1:m){
      if(x[i,j] >= 0){
        count <- count + 1
      }
    }
  }

  return(count)
}

matrix_1 = matrix(c(1, -1, 3, 4, 5, 6, 7, 8, 9), nrow=3, ncol=3, byrow=TRUE)
row_wise_scan(matrix_1)
```

[1] 8

```
#my_vec_bool <- sum((ifelse(my_vec_double <= 0, TRUE, FALSE))==FALSE)
```

Part 3

```
col_wise_scan <- function(x){
  count <- 0

  # Insert your code here
  n <- nrow(x)
  m <- ncol(x)
  for(j in 1:m){
    for(i in 1:n){
      if(x[i,j] >= 0){
        count <- count + 1
      }
    }
  }
}
```

```

    }
}

    return(count)
}

matrix_1 = matrix(c(1, -1, 3, 4, 5, 6, 7, 8, 9), nrow=3, ncol=3, byrow=TRUE)
col_wise_scan(matrix_1)

```

```
[1] 8
```

Part 4

"col_wise_scan and row_wise_scan will have similar runtime. Both codes traverse through ea

```
[1] "col_wise_scan and row_wise_scan will have similar runtime. Both codes traverse through e
```

Part 5

```

time_scan <- function(f, M){
  initial_time <- Sys.time() # Write your code here
  f(M)
  final_time <- Sys.time() # Write your code here

  total_time_taken <- final_time - initial_time
  return(total_time_taken)
}

M = matrix(c((1:100)**2), nrow=1, ncol=100)

list(
  row_wise_time = time_scan(row_wise_scan, M),
  col_wise_time = time_scan(col_wise_scan, M)
)

```

```
$row_wise_time  
Time difference of 0 secs
```

```
$col_wise_time  
Time difference of 0.0003881454 secs
```

Part 6

```
M<-matrix(rnorm(25) , nrow = 100, ncol=100)  
  
list(  
  row_wise_time = time_scan(row_wise_scan, M),  
  col_wise_time = time_scan(col_wise_scan, M)  
)
```

```
$row_wise_time  
Time difference of 0.0007071495 secs
```

```
$col_wise_time  
Time difference of 0.00198698 secs
```

```
M<-matrix(rnorm(25) , nrow = 1000, ncol=1000)  
  
list(  
  row_wise_time = time_scan(row_wise_scan, M),  
  col_wise_time = time_scan(col_wise_scan, M)  
)
```

```
$row_wise_time  
Time difference of 0.1588171 secs
```

```
$col_wise_time  
Time difference of 0.1446559 secs
```

```
M<-matrix(rnorm(25) , nrow = 5000, ncol=5000)  
  
list(  
  row_wise_time = time_scan(row_wise_scan, M),  
  col_wise_time = time_scan(col_wise_scan, M)
```

```
)
```

```
$row_wise_time
```

```
Time difference of 3.969174 secs
```

```
$col_wise_time
```

```
Time difference of 3.181715 secs
```

```
"seems like as the number of rows and columns increase, the runtime of both algorithms tak
```

```
[1] "seems like as the number of rows and columns increase, the runtime of both algorithms t
```