HW 1

Rosalind Shui

Question 2

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

1.

```r
typeof(my_vec)
```

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

2.

```r
my_vec_double <- as.double(my_vec)
my_vec_int <- as.integer(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
```

```r
my_vec_int
```

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

3.

```r
my_vec_bool <- as.logical(my_vec_double >= 0)

my_vec_bool
```

```
 [1]  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE
```

Four elements are greater than zero

4.

```r
sort(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

1.

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

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

```r
a <- c(1:100)
b <- c(a^2)
```

```r
matrix(
  c(a, b),
  nrow = 2,
  byrow = TRUE
)
```

```
     [,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
```

2.

```r
generate_matrix <- function(n){
  return(
    matrix(
      rnorm(n^2),
      nrow = n
    )
  )
}

M <- generate_matrix(50)
```

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

  count <- 0
  for(row in 1:n){
    for(col in 1:m){
      if(M[row,col] >= 0){
        count <- count + 1
      }
    }
  }
  return(count)
}

row_wise_scan(M)
```

```
[1] 1230
```

3.

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

  count <- 0
  for(col in 1:m){
    for(row in 1:n){
      if(x[col,row] >= 0){
        count <- count + 1
      }
    }
  }
  return(count)
}
```

```r
col_wise_scan(M)
```

```
[1] 1230
```

4. I expect `row_wise_scan` to take longer because R takes longer to read wide data than long data.

5.

```r
time_scan <- function(f, M){
  initial_time <- Sys.time()
  f(M)
  final_time <- Sys.time()

  total_time_taken <- final_time - initial_time
  return(total_time_taken)
}

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 0.0009958744 secs

$col_wise_time
Time difference of 0 secs
```

`col_wise_time` took more time to run.

6.

```r
M <- generate_matrix(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 secs
```

```r
M <- generate_matrix(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 0.1326449 secs

$col_wise_time
Time difference of 0.1057172 secs
```

```r
M <- generate_matrix(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 3.161521 secs

$col_wise_time
Time difference of 2.277908 secs
```

`col_wise_scan` takes less time to run on larger matrices.

Appendix

```r
sessionInfo()
```

```
R version 4.1.2 (2021-11-01)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19045)
```

```
Matrix products: default

locale:
[1] LC_COLLATE=English_United States.1252
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] stats     graphics  grDevices datasets  utils     methods   base

loaded via a namespace (and not attached):
 [1] compiler_4.1.2  fastmap_1.1.0   cli_3.6.0       htmltools_0.5.4
 [5] tools_4.1.2     yaml_2.3.7      rmarkdown_2.20  knitr_1.42
 [9] xfun_0.36       digest_0.6.31   jsonlite_1.8.4  rlang_1.0.6
[13] renv_0.16.0-53  evaluate_0.20
```