

Lab 2

Connor Brown

Basic R Skills

First, install the package `testthat` (a widely accepted testing suite for R) from <https://github.com/r-lib/testthat> using `pacman`. If you are using Windows, this will be a long install, but you have to go through it for some of the stuff we are doing in class. LINUX (or MAC) is preferred for coding. If you can't get it to work, install this package from CRAN (still using `pacman`), but this is not recommended long term.

```
if (!require("pacman")){install.packages("pacman")} #installs pacman if necessary but does not load it!
```

```
## Loading required package: pacman
```

```
pacman::p_load(testthat)
```

- Use the `seq` function to create vector `v` consisting of all numbers from -100 to 100.

```
v = seq(-100, 100)
```

```
v
```

```
##      [1] -100 -99 -98 -97 -96 -95 -94 -93 -92 -91 -90 -89 -88 -87
##      [15] -86 -85 -84 -83 -82 -81 -80 -79 -78 -77 -76 -75 -74 -73
##      [29] -72 -71 -70 -69 -68 -67 -66 -65 -64 -63 -62 -61 -60 -59
##      [43] -58 -57 -56 -55 -54 -53 -52 -51 -50 -49 -48 -47 -46 -45
##      [57] -44 -43 -42 -41 -40 -39 -38 -37 -36 -35 -34 -33 -32 -31
##      [71] -30 -29 -28 -27 -26 -25 -24 -23 -22 -21 -20 -19 -18 -17
##      [85] -16 -15 -14 -13 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3
##      [99] -2 -1  0  1  2  3  4  5  6  7  8  9 10 11
##     [113] 12 13 14 15 16 17 18 19 20 21 22 23 24 25
##     [127] 26 27 28 29 30 31 32 33 34 35 36 37 38 39
##     [141] 40 41 42 43 44 45 46 47 48 49 50 51 52 53
##     [155] 54 55 56 57 58 59 60 61 62 63 64 65 66 67
##     [169] 68 69 70 71 72 73 74 75 76 77 78 79 80 81
##     [183] 82 83 84 85 86 87 88 89 90 91 92 93 94 95
##     [197] 96 97 98 99 100
```

Test using the following code:

```
expect_equal(v, -100 : 100)
```

If there are any errors, the `expect_equal` function will tell you about them. If there are no errors, then it will be silent.

- Create a function `my_reverse` which takes as required input a vector and returns the vector in reverse where the first entry is the last entry, etc. No function calls are allowed inside your function (otherwise that would defeat the purpose of the exercise).

```
my_reverse = function(inputVector){
```

```
  #Makes empty vector the same size as input vector
```

```
  reverseVector = rep(NA, length(inputVector))
```

```
  #Loops from 1 : size of input vector
```

```
  for (i in seq(1, length(inputVector))) {
```

```
    #iterates up in new vector, iterates down in input vector
```

```

    reverseVector[i] = inputVector[(length(inputVector)+1) - i]
  }

  #returns new vector
  reverseVector
}

```

Test using the following code:

```

expect_equal(my_reverse(c("A", "B", "C")), c("C", "B", "A"))
expect_equal(my_reverse(v), rev(v))

```

- Let $n = 50$. Create a $n \times n$ matrix R of exactly 50% entries 0's, 25% 1's 25% 2's in random locations.

```

n = 50

#Creates vector w/50% entries 0's, 25% 1's, 25% 2's randomly ordered
f = sample(c(rep(0, 0.5*(n*n)), rep(1, 0.25*(n*n)), rep(2, 0.25*(n*n))))

```

```

#Creates matrix with f vector as input for n*n matrix
R = matrix(f, nrow = n, ncol = n)
R

```

```

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]  0   0   2   0   0   0   0   0   0   1   0   0   2
## [2,]  1   1   0   0   0   0   2   0   1   0   1   1   2
## [3,]  1   0   0   2   1   0   0   0   2   0   0   0   0
## [4,]  2   0   2   0   0   1   1   0   1   0   0   2   0
## [5,]  2   0   2   0   2   2   1   1   2   1   1   0   1
## [6,]  0   0   0   0   0   1   0   0   1   0   1   1   2
## [7,]  2   2   2   2   0   0   1   0   0   2   0   1   0
## [8,]  2   2   2   0   1   0   2   1   1   1   0   2   2
## [9,]  0   1   1   2   2   2   2   1   1   2   0   2   0
## [10,] 1   0   2   1   1   2   2   2   0   0   1   2   0
## [11,] 0   2   0   0   0   0   0   0   1   0   0   0   0
## [12,] 0   0   2   1   1   1   1   2   0   1   1   1   0
## [13,] 2   0   2   0   1   1   0   1   2   0   0   0   0
## [14,] 1   0   0   0   2   1   0   2   1   2   2   2   0
## [15,] 0   1   2   0   1   2   1   2   1   1   2   2   2
## [16,] 2   1   2   2   1   1   2   2   2   0   0   2   0
## [17,] 2   2   1   1   0   0   0   0   0   0   0   2   1
## [18,] 1   0   2   1   0   0   0   0   1   0   2   2   0
## [19,] 0   2   1   2   0   0   0   2   1   0   1   0   0
## [20,] 0   0   0   0   2   0   2   0   0   0   1   2   0
## [21,] 0   0   1   0   0   1   1   0   0   1   0   0   2
## [22,] 0   2   2   0   0   1   1   0   0   2   1   0   2
## [23,] 0   2   1   0   0   0   1   0   0   2   1   2   0
## [24,] 0   2   1   0   0   0   0   0   1   2   0   2   0
## [25,] 2   0   0   0   2   0   1   1   1   0   0   2   0
## [26,] 2   0   1   1   0   2   0   0   2   0   2   0   1
## [27,] 0   0   1   1   0   0   0   1   0   1   0   0   2
## [28,] 1   1   1   0   2   0   2   0   0   0   2   2   2
## [29,] 1   0   1   2   1   2   0   0   0   2   1   1   1
## [30,] 0   0   0   1   1   1   1   0   0   0   0   2   0
## [31,] 0   0   2   1   0   2   2   1   1   2   2   0   0

```

## [32,]	0	0	2	1	0	0	0	1	2	0	0	2	0
## [33,]	2	0	1	0	0	0	1	0	0	2	2	0	2
## [34,]	0	1	2	2	0	0	1	0	1	0	0	0	1
## [35,]	2	1	0	0	0	2	0	1	0	0	1	2	0
## [36,]	0	0	2	0	1	2	0	2	0	1	0	2	1
## [37,]	2	1	2	1	0	2	0	0	0	0	0	0	0
## [38,]	0	0	1	0	2	0	1	2	0	0	0	0	0
## [39,]	2	2	1	2	2	1	2	1	2	1	2	0	0
## [40,]	0	0	1	0	1	2	0	0	1	0	1	0	0
## [41,]	2	0	0	1	1	2	2	0	2	2	0	2	1
## [42,]	0	1	0	0	2	0	0	0	0	0	1	1	0
## [43,]	2	0	0	2	0	1	0	0	1	0	1	1	1
## [44,]	2	0	0	1	0	1	0	1	1	1	1	2	2
## [45,]	1	2	1	0	0	0	1	0	0	0	1	0	1
## [46,]	1	0	2	0	1	0	2	0	0	0	1	0	1
## [47,]	1	1	0	0	0	2	2	2	0	2	2	1	0
## [48,]	2	0	0	1	0	1	0	1	0	0	0	0	1
## [49,]	0	0	2	2	0	1	0	0	0	2	0	2	1
## [50,]	0	0	0	0	2	1	0	0	0	1	2	2	1
##	[,14]	[,15]	[,16]	[,17]	[,18]	[,19]	[,20]	[,21]	[,22]	[,23]	[,24]		
## [1,]	0	1	2	0	1	1	0	2	2	0	1		
## [2,]	0	0	1	1	1	0	2	2	2	0	0		
## [3,]	0	1	0	0	1	0	0	2	1	0	0		
## [4,]	0	1	0	2	0	0	0	0	0	0	2		
## [5,]	1	0	1	0	1	2	2	1	1	0	2		
## [6,]	0	0	0	1	2	2	0	0	1	0	0		
## [7,]	1	0	0	2	0	2	2	2	1	2	0		
## [8,]	0	0	0	0	0	1	2	0	1	0	0		
## [9,]	1	0	1	1	0	1	1	0	1	0	2		
## [10,]	1	0	0	1	1	0	1	2	0	1	0		
## [11,]	2	2	1	0	0	0	0	0	1	2	0		
## [12,]	0	0	0	2	2	1	2	0	2	2	2		
## [13,]	2	0	0	0	2	2	0	0	2	1	0		
## [14,]	2	1	2	2	2	0	2	0	2	1	1		
## [15,]	2	1	0	0	1	2	0	0	2	2	1		
## [16,]	0	0	1	0	2	0	0	0	2	0	2		
## [17,]	0	0	0	0	0	0	1	2	0	0	1		
## [18,]	0	0	2	1	2	2	0	0	0	0	2		
## [19,]	0	2	0	0	0	0	2	0	1	0	1		
## [20,]	0	0	0	2	1	0	0	2	1	0	0		
## [21,]	1	2	2	0	2	2	2	0	0	0	2		
## [22,]	2	1	2	1	0	1	0	1	0	0	0		
## [23,]	2	1	2	2	1	2	0	0	0	2	2		
## [24,]	0	0	0	2	0	0	2	0	2	0	0		
## [25,]	2	1	0	0	0	0	0	0	0	0	0		
## [26,]	2	0	0	0	1	0	0	1	1	0	0		
## [27,]	0	2	0	0	2	2	2	0	1	2	2		
## [28,]	2	0	1	1	0	0	0	0	0	0	0		
## [29,]	0	2	0	2	2	2	0	2	0	0	0		
## [30,]	2	0	1	2	1	0	2	1	0	1	0		
## [31,]	1	1	0	0	2	2	1	1	1	2	2		
## [32,]	0	2	0	2	2	0	1	0	2	0	2		
## [33,]	0	2	0	0	0	0	2	2	0	0	0		
## [34,]	0	0	0	0	0	2	0	2	0	1	2		

## [35,]	0	0	2	2	0	2	0	0	0	1	1
## [36,]	0	0	0	0	1	2	1	1	1	0	2
## [37,]	2	2	0	0	0	0	1	0	2	0	0
## [38,]	0	2	2	0	1	0	0	1	0	1	2
## [39,]	0	1	1	0	0	0	2	0	2	2	0
## [40,]	2	1	1	2	1	1	2	1	0	2	0
## [41,]	0	0	0	1	2	0	0	2	2	0	0
## [42,]	1	2	0	1	0	0	1	0	0	1	0
## [43,]	0	0	0	0	0	1	0	0	0	0	0
## [44,]	1	1	1	0	2	0	0	2	0	0	1
## [45,]	0	0	0	2	2	0	0	0	1	2	0
## [46,]	0	2	1	0	0	1	0	2	2	2	0
## [47,]	0	1	0	2	0	0	2	2	2	0	2
## [48,]	1	2	1	0	0	0	0	0	0	1	2
## [49,]	0	1	0	2	0	2	0	0	0	0	0
## [50,]	0	2	2	0	1	0	0	0	2	1	0
##	[,25]	[,26]	[,27]	[,28]	[,29]	[,30]	[,31]	[,32]	[,33]	[,34]	[,35]
## [1,]	0	0	0	0	0	0	0	0	0	2	0
## [2,]	2	0	1	2	1	0	0	0	1	2	0
## [3,]	2	0	2	0	1	0	0	1	0	2	0
## [4,]	1	1	0	2	0	0	1	0	0	0	0
## [5,]	1	2	2	0	2	0	2	2	0	0	0
## [6,]	0	0	1	0	0	0	0	2	2	1	2
## [7,]	0	0	2	1	2	2	0	0	0	0	0
## [8,]	2	0	0	2	1	2	0	1	0	2	0
## [9,]	0	0	0	0	0	2	0	2	1	0	1
## [10,]	1	1	0	2	0	1	0	1	2	0	2
## [11,]	1	0	2	0	0	0	1	0	1	0	0
## [12,]	0	1	1	0	0	0	1	0	2	0	2
## [13,]	1	0	1	1	1	1	2	1	1	1	2
## [14,]	2	1	0	0	1	1	0	2	0	2	0
## [15,]	2	0	0	2	2	2	1	0	2	0	2
## [16,]	2	0	0	1	2	0	1	0	2	2	0
## [17,]	2	0	2	0	0	1	1	0	0	0	1
## [18,]	0	1	2	0	2	0	0	0	2	0	0
## [19,]	1	0	2	1	1	1	1	1	0	1	1
## [20,]	1	0	0	0	0	0	2	1	0	0	1
## [21,]	2	0	2	0	0	0	0	0	0	0	1
## [22,]	0	0	0	0	1	0	1	0	0	0	2
## [23,]	0	0	0	2	1	0	1	0	1	0	1
## [24,]	0	1	0	0	0	0	2	0	1	0	1
## [25,]	0	0	1	2	0	1	2	0	1	0	1
## [26,]	1	0	1	1	2	0	0	1	1	2	1
## [27,]	0	2	0	1	0	0	0	0	2	2	0
## [28,]	2	1	0	1	2	1	0	1	0	0	2
## [29,]	0	0	2	2	1	1	0	0	1	2	0
## [30,]	2	0	0	0	2	2	0	0	1	0	0
## [31,]	2	0	0	0	0	1	1	1	0	2	0
## [32,]	0	0	1	2	0	0	0	1	2	0	2
## [33,]	1	0	0	0	1	0	0	0	0	1	0
## [34,]	2	0	2	1	1	1	0	1	2	0	0
## [35,]	0	0	0	0	2	0	0	2	1	0	1
## [36,]	1	0	2	1	1	0	0	0	0	1	0
## [37,]	0	2	0	1	0	0	2	2	0	1	0

## [38,]	0	2	1	2	0	0	0	2	0	0	1
## [39,]	0	1	2	0	2	2	0	0	2	0	0
## [40,]	0	0	1	2	2	0	1	0	2	1	0
## [41,]	0	1	2	0	0	2	1	1	2	1	2
## [42,]	1	0	0	0	0	0	1	0	0	0	0
## [43,]	0	1	2	0	0	2	0	2	0	1	0
## [44,]	2	1	0	0	0	0	2	1	0	1	0
## [45,]	0	2	0	0	0	0	2	0	0	2	0
## [46,]	2	2	1	0	2	0	1	2	1	2	2
## [47,]	0	1	1	2	0	2	2	2	0	2	1
## [48,]	0	0	0	1	0	0	0	2	1	0	0
## [49,]	2	1	0	0	0	0	0	0	1	1	1
## [50,]	2	0	0	0	0	0	1	0	0	0	1
##	[,36]	[,37]	[,38]	[,39]	[,40]	[,41]	[,42]	[,43]	[,44]	[,45]	[,46]
## [1,]	0	0	0	0	2	0	0	0	0	0	0
## [2,]	0	1	2	0	2	0	0	1	0	1	1
## [3,]	2	0	2	0	0	2	2	0	1	0	1
## [4,]	0	2	1	0	0	1	2	1	2	1	2
## [5,]	1	2	2	1	0	2	0	1	0	1	2
## [6,]	0	2	0	0	2	1	0	0	2	2	0
## [7,]	1	0	1	2	2	2	2	0	2	1	2
## [8,]	0	0	1	1	1	1	0	1	1	0	1
## [9,]	0	0	2	1	2	0	0	1	0	0	0
## [10,]	1	0	2	2	1	1	0	0	2	0	0
## [11,]	2	2	2	1	1	1	2	1	1	0	2
## [12,]	0	2	2	0	1	2	0	0	0	0	2
## [13,]	0	2	0	0	2	0	0	0	0	2	0
## [14,]	0	2	2	0	2	0	0	0	1	0	0
## [15,]	2	1	0	0	1	0	2	0	1	0	0
## [16,]	2	0	0	0	0	0	2	1	1	0	1
## [17,]	0	1	0	2	1	0	1	1	2	2	0
## [18,]	2	1	0	0	0	1	2	1	2	1	0
## [19,]	1	0	1	0	1	0	2	1	0	2	1
## [20,]	0	2	2	1	1	2	0	2	1	0	0
## [21,]	0	0	2	0	1	0	1	0	1	0	0
## [22,]	0	0	0	2	2	0	2	2	1	2	0
## [23,]	0	1	0	2	1	0	1	1	1	0	1
## [24,]	1	2	0	1	0	0	2	1	0	0	2
## [25,]	0	1	2	0	1	0	1	1	0	0	2
## [26,]	0	0	0	1	0	0	2	0	0	0	0
## [27,]	2	0	1	1	1	0	1	2	1	0	0
## [28,]	0	2	0	0	2	0	2	2	1	0	1
## [29,]	1	1	0	2	1	1	0	1	2	2	2
## [30,]	2	0	1	0	1	2	1	1	1	0	2
## [31,]	1	0	0	0	0	0	2	0	2	1	0
## [32,]	0	2	2	1	0	2	0	1	0	0	2
## [33,]	0	0	0	0	0	2	0	1	2	0	2
## [34,]	0	1	2	0	1	1	0	0	1	0	1
## [35,]	0	2	1	2	1	0	2	2	2	2	0
## [36,]	0	2	0	2	0	0	1	1	2	0	1
## [37,]	0	0	0	2	0	2	2	2	0	1	2
## [38,]	2	0	0	0	0	2	0	1	0	0	2
## [39,]	0	0	0	0	0	2	0	0	2	0	2
## [40,]	1	0	0	0	2	1	0	1	0	1	1

## [41,]	2	0	0	0	2	0	1	1	1	0	0
## [42,]	1	1	2	1	2	1	0	0	0	0	0
## [43,]	1	1	0	1	0	2	0	0	1	0	1
## [44,]	1	1	0	2	0	0	0	0	0	0	2
## [45,]	1	0	2	1	0	1	0	2	2	0	1
## [46,]	0	1	2	0	0	0	2	0	0	2	0
## [47,]	0	0	1	0	0	2	0	1	0	1	2
## [48,]	1	1	1	2	0	1	1	0	1	1	1
## [49,]	0	0	0	0	2	2	1	0	0	1	1
## [50,]	1	0	2	1	0	2	0	1	0	0	0
##	[,47]	[,48]	[,49]	[,50]							
## [1,]	0	2	1	2							
## [2,]	1	0	2	0							
## [3,]	1	0	0	2							
## [4,]	2	1	0	2							
## [5,]	0	1	2	2							
## [6,]	0	2	0	1							
## [7,]	0	1	2	2							
## [8,]	0	1	0	1							
## [9,]	0	1	2	1							
## [10,]	0	1	0	1							
## [11,]	2	0	2	0							
## [12,]	0	2	0	0							
## [13,]	0	2	2	0							
## [14,]	0	0	1	1							
## [15,]	0	0	1	1							
## [16,]	1	2	0	1							
## [17,]	0	0	1	1							
## [18,]	1	0	0	2							
## [19,]	2	1	0	1							
## [20,]	0	0	1	1							
## [21,]	0	1	0	2							
## [22,]	0	0	0	0							
## [23,]	0	0	0	0							
## [24,]	2	0	2	2							
## [25,]	0	0	1	2							
## [26,]	1	0	2	0							
## [27,]	1	2	0	1							
## [28,]	2	0	1	1							
## [29,]	2	0	0	0							
## [30,]	2	1	0	1							
## [31,]	0	1	1	1							
## [32,]	0	2	1	2							
## [33,]	0	2	2	0							
## [34,]	1	0	0	0							
## [35,]	0	2	1	0							
## [36,]	0	0	1	0							
## [37,]	1	1	2	2							
## [38,]	2	1	1	1							
## [39,]	2	0	2	0							
## [40,]	1	1	0	2							
## [41,]	0	1	2	0							
## [42,]	0	1	1	0							
## [43,]	2	2	1	2							

```
## [44,] 2 1 0 2
## [45,] 2 0 0 1
## [46,] 1 0 0 0
## [47,] 0 0 0 2
## [48,] 0 0 0 0
## [49,] 1 2 0 0
## [50,] 0 0 2 0
```

Test using the following and write two more tests as specified below:

```
expect_equal(dim(R), c(n, n))

#Test that the only unique values are 0, 1, 2

#Vectorizes the matrix R
vector_R = c(R)
#Passes unique values in R to the vector
unique_R_values = sort(unique(vector_R))
#Tests to see if the 2 vectors are equal
expect_equal(unique_R_values, c(0, 1, 2))

#Test that there are exactly 625 2's

#Makes table of number of unique values in R
summary_of_R = table(R)
summary_of_R

## R
## 0 1 2
## 1250 625 625

#Converts table to dataframe
summary_of_R_asDF = data.frame(summary_of_R)
summary_of_R_asDF

## R Freq
## 1 0 1250
## 2 1 625
## 3 2 625

#Passes # of 2's into variable, "num_2s"
num_2s = summary_of_R_asDF$Freq[2]

expect_equal(num_2s, 625)
```

- Randomly punch holes (i.e. NA) values in this matrix so that approximately 30% of the entries are missing.

```
#Creates vector w/~30% 1's
NA_Vector = rbinom(n*n, 1, prob = .3)
NA_Vector

## [1] 1 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 1 0 1 0 0 1 1 0 0 1
## [35] 0 1 0 0 0 1 1 0 0 0 0 1 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0
## [69] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 1 0 0 0 0 1 0 0 0 0 0 0
```

[illegible]


```
## [1939] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0
## [1973] 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0
## [2007] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0
## [2041] 0 1 0 0 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 1 1 0 0 0 0
## [2075] 1 1 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0
## [2109] 0 1 1 0 1 0 1 1 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 1
## [2143] 0 0 0 0 0 1 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 1 0
## [2177] 0 0 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0
## [2211] 1 0 1 0 1 0 0 1 1 1 0 0 0 1 0 0 0 0 0 1 0 0 1 1 1 0 1 0 0 0 0 0 1 1
## [2245] 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 0 1 0 1 1 0 0 1 0 1 1 0 0 0 0 1 1 0 1
## [2279] 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 1 0 1 0 0 0 0 1 1 1 0 0 1 0
## [2313] 0 0 0 0 0 0 1 1 0 0 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1
## [2347] 0 1 1 0 1 0 1 0 1 0 0 0 0 0 1 1 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1
## [2381] 1 0 0 1 0 0 1 0 1 1 0 1 1 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 0 0 0 1 0 1
## [2415] 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 0 1 0 1 1
## [2449] 1 0 0 0 1 1 0 1 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0
## [2483] 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1
```

```
#Replaces the 1's in NA_Vector w/NA's, fills the rest of the vector w/values from the original R, and m
R = matrix(ifelse(NA_Vector == 1, NA, R), n, n)
R
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]  NA   0   2   0   0   0   0   NA   NA   NA   0   0   NA
## [2,]   1  NA   0   0  NA   0  NA   0   NA   0   1   1   2
## [3,]   1  NA   0   2   1   0   0   0   2   0   0   0   0
## [4,]  NA   0   2  NA  NA   1   1   0   1   0  NA   2   0
## [5,]   2   0   2   0   2  NA   1  NA   2  NA   1   0   1
## [6,]   0  NA   0   0  NA   1  NA   0   1  NA   1   1   2
## [7,]  NA   2  NA   2   0   0   1   0  NA   2  NA   1   0
## [8,]   2  NA   2  NA   1   0   2   1   1   1   0   2   2
## [9,]   0   1  NA  NA  NA  NA   2  NA   1   2   0   2  NA
## [10,]  1   0   2  NA   1   2   2  NA  NA   0   1   2  NA
## [11,]  0  NA   0   0   0   0   0   0  NA  NA   0   0   0
## [12,]  0   0  NA  NA   1   1   1   2  NA   1   1  NA   0
## [13,]  NA   0   2   0  NA   1   0  NA   2  NA   0   0   0
## [14,]   1   0   0  NA   2   1   0  NA   1   2   2  NA   0
## [15,]  NA   1   2  NA   1   2  NA  NA   1   1   2   2   2
## [16,]   2   1   2   2   1   1  NA   2   2   0   0  NA   0
## [17,]   2   2   1   1   0   0   0   0   0   0   0   2   1
## [18,]   1   0   2   1   0   0   0   0   1  NA   2  NA   0
## [19,]   0   2   1   2   0  NA   0   2  NA   0   1  NA   0
## [20,]   0   0   0   0   2   0   2   0  NA  NA   1   2   0
## [21,]   0   0   1   0   0  NA   1   0   0   1   0   0   2
## [22,]   0  NA   2   0  NA   1  NA  NA  NA  NA  NA   0   2
## [23,]  NA   2   1   0   0  NA  NA  NA   0   2   1   2   0
## [24,]   0   2   1   0  NA  NA   0   0   1   2   0  NA  NA
## [25,]  NA   0   0  NA   2   0   1   1   1   0  NA   2  NA
## [26,]   2   0   1  NA   0   2   0  NA   2   0   2   0  NA
## [27,]  NA   0  NA   1   0  NA   0  NA  NA   1   0   0   2
## [28,]   1   1   1  NA   2   0   2  NA  NA   0   2  NA   2
## [29,]   1   0  NA   2  NA   2   0   0   0   2   1   1   1
## [30,]  NA   0   0   1   1   1  NA   0  NA   0   0   2  NA
## [31,]  NA   0  NA   1   0   2   2   1  NA   2   2   0   0
## [32,]   0  NA   2   1  NA   0   0  NA   2   0  NA  NA   0
```

## [33,]	2	0	NA	0	0	NA	1	0	NA	2	2	NA	2
## [34,]	NA	1	2	2	0	0	1	0	1	NA	0	0	NA
## [35,]	2	1	0	NA	NA	2	0	1	NA	NA	1	2	0
## [36,]	NA	0	2	NA	1	2	0	2	0	1	NA	NA	1
## [37,]	2	NA	2	NA	0	2	0	NA	NA	0	0	0	NA
## [38,]	0	NA	1	0	NA	0	NA	2	0	0	0	0	NA
## [39,]	2	2	1	2	2	1	2	1	2	1	2	0	0
## [40,]	NA	NA	1	NA	1	2	0	0	1	0	NA	0	0
## [41,]	NA	NA	NA	1	1	2	2	NA	2	NA	0	NA	1
## [42,]	0	1	NA	NA	NA	0	NA	0	0	0	1	NA	0
## [43,]	2	0	0	2	NA	NA	0	NA	1	NA	1	1	NA
## [44,]	2	0	0	1	0	1	0	NA	1	NA	1	2	2
## [45,]	1	2	1	0	0	0	NA	0	NA	0	NA	0	NA
## [46,]	NA	NA	2	NA	1	0	NA	0	NA	0	NA	NA	NA
## [47,]	1	1	NA	NA	NA	2	2	2	0	2	2	NA	NA
## [48,]	NA	0	NA	1	NA	NA	0	NA	NA	0	0	NA	1
## [49,]	NA	0	NA	NA	0	NA	NA	0	0	2	NA	2	1
## [50,]	0	0	0	0	NA	1	0	0	0	1	2	2	1
##	[,14]	[,15]	[,16]	[,17]	[,18]	[,19]	[,20]	[,21]	[,22]	[,23]	[,24]		
## [1,]	NA	1	2	0	1	1	0	2	2	0	1		
## [2,]	NA	NA	1	NA	1	NA	NA	2	NA	0	NA		
## [3,]	0	1	0	0	1	0	NA	2	1	0	0		
## [4,]	0	1	0	2	0	0	NA	0	0	NA	2		
## [5,]	1	0	1	NA	1	2	2	1	1	NA	NA		
## [6,]	0	NA	0	1	2	NA	NA	0	1	0	0		
## [7,]	NA	0	NA	NA	0	NA	NA	2	1	2	NA		
## [8,]	NA	NA	0	0	NA	NA	2	NA	1	NA	0		
## [9,]	1	0	1	1	0	1	NA	NA	NA	0	2		
## [10,]	NA	0	0	1	NA	0	1	2	0	NA	0		
## [11,]	2	2	1	0	0	0	0	NA	1	2	0		
## [12,]	0	NA	0	2	2	1	NA	0	2	2	2		
## [13,]	2	0	0	NA	2	NA	0	NA	2	1	0		
## [14,]	2	1	2	2	2	0	2	0	NA	1	1		
## [15,]	2	1	0	0	1	NA	0	NA	2	2	1		
## [16,]	NA	0	1	NA	2	0	0	NA	2	NA	2		
## [17,]	NA	0	0	0	0	0	1	2	0	0	NA		
## [18,]	0	NA	2	1	NA	2	0	NA	0	0	2		
## [19,]	0	2	NA	0	0	NA	2	0	1	0	1		
## [20,]	0	0	NA	NA	1	0	0	NA	1	0	0		
## [21,]	NA	NA	NA	0	NA	NA	NA	0	NA	NA	NA		
## [22,]	2	NA	2	1	NA	1	NA	NA	0	0	0		
## [23,]	2	NA	2	2	NA	2	NA	0	0	2	2		
## [24,]	0	NA	0	2	0	NA	2	NA	NA	0	0		
## [25,]	NA	1	0	NA	NA	0	0	0	0	0	0		
## [26,]	2	0	0	0	1	0	NA	1	NA	0	0		
## [27,]	NA	2	0	NA	NA	NA	2	0	1	2	NA		
## [28,]	2	0	NA	1	0	NA	0	0	0	NA	0		
## [29,]	0	NA	0	2	2	2	NA	NA	NA	0	0		
## [30,]	NA	NA	NA	2	1	0	2	NA	0	NA	0		
## [31,]	1	NA	0	0	NA	2	NA	NA	1	NA	2		
## [32,]	0	2	0	NA	NA	0	1	NA	NA	0	2		
## [33,]	0	2	0	NA	NA	0	2	NA	0	0	0		
## [34,]	0	0	0	NA	0	2	0	2	0	1	2		
## [35,]	0	0	2	2	0	2	0	0	NA	1	1		

## [36,]	0	0	NA	0	1	2	1	1	NA	0	2
## [37,]	2	2	0	0	0	NA	NA	0	2	NA	0
## [38,]	0	2	2	0	1	0	NA	NA	NA	1	2
## [39,]	NA	1	1	0	NA	0	NA	0	2	NA	NA
## [40,]	2	1	NA	2	NA	1	NA	1	0	NA	0
## [41,]	0	0	NA	1	2	0	0	2	NA	NA	0
## [42,]	1	2	0	1	NA	0	NA	NA	NA	NA	NA
## [43,]	0	0	0	0	0	NA	0	0	NA	0	NA
## [44,]	1	1	1	0	2	0	NA	2	0	0	NA
## [45,]	0	0	0	2	2	NA	0	0	NA	2	0
## [46,]	0	NA	NA	0	NA	1	0	2	2	2	0
## [47,]	NA	1	NA	2	0	NA	NA	NA	2	0	NA
## [48,]	NA	2	1	0	NA	NA	0	0	0	1	2
## [49,]	0	NA	0	NA	0	NA	0	0	0	NA	0
## [50,]	0	2	2	NA	1	0	NA	0	NA	1	0
##	[,25]	[,26]	[,27]	[,28]	[,29]	[,30]	[,31]	[,32]	[,33]	[,34]	[,35]
## [1,]	0	0	NA	0	NA	0	0	NA	0	2	0
## [2,]	2	NA	1	NA	1	0	0	0	1	2	0
## [3,]	2	0	NA	0	NA	0	NA	1	NA	2	0
## [4,]	1	NA	0	2	0	0	NA	0	NA	NA	0
## [5,]	1	2	NA	0	2	0	2	2	0	NA	NA
## [6,]	NA	0	1	0	NA	0	0	2	2	1	2
## [7,]	0	0	NA	NA	2	NA	0	0	0	NA	0
## [8,]	NA	NA	0	NA	1	2	NA	1	0	2	0
## [9,]	0	0	0	NA	0	2	0	2	1	0	NA
## [10,]	1	1	0	2	0	1	0	1	NA	0	2
## [11,]	1	0	2	NA	0	NA	1	0	1	NA	0
## [12,]	0	NA	1	0	0	0	1	NA	2	0	NA
## [13,]	1	0	1	1	1	1	NA	1	1	1	2
## [14,]	NA	1	NA	0	NA	1	NA	2	0	2	NA
## [15,]	2	NA	NA	2	2	2	1	0	NA	0	2
## [16,]	2	NA	NA	1	2	NA	1	0	NA	2	0
## [17,]	NA	0	2	NA	0	NA	1	NA	NA	0	1
## [18,]	0	NA	2	NA	2	0	0	0	NA	0	0
## [19,]	1	0	2	1	1	1	1	1	0	NA	NA
## [20,]	NA	0	0	0	0	0	NA	NA	0	NA	1
## [21,]	2	0	2	NA	NA	0	0	0	0	NA	1
## [22,]	0	0	0	0	1	0	1	0	0	NA	2
## [23,]	0	0	0	2	1	NA	1	0	1	0	1
## [24,]	0	1	0	NA	0	0	2	0	NA	0	NA
## [25,]	0	0	1	2	0	1	NA	0	1	0	1
## [26,]	1	0	NA	NA	2	0	0	NA	1	2	1
## [27,]	0	2	0	NA	0	0	0	0	NA	2	0
## [28,]	2	1	0	1	NA	1	0	1	0	0	2
## [29,]	NA	0	NA	NA	1	1	0	NA	1	2	0
## [30,]	2	NA	0	0	2	2	0	NA	1	NA	NA
## [31,]	NA	0	0	NA	0	1	1	1	NA	2	NA
## [32,]	0	NA	1	2	NA	0	0	NA	2	NA	2
## [33,]	1	0	NA	0	1	0	NA	0	NA	NA	NA
## [34,]	2	NA	2	1	1	1	0	1	2	0	0
## [35,]	NA	0	0	0	2	0	0	2	1	0	NA
## [36,]	NA	0	2	1	1	0	0	0	NA	1	NA
## [37,]	NA	2	NA	NA	NA	NA	2	2	0	NA	0
## [38,]	0	2	1	NA	NA	NA	0	NA	0	NA	1

##	[39,]	NA	1	2	0	2	2	NA	0	2	0	NA
##	[40,]	NA	0	1	2	2	NA	1	0	2	1	0
##	[41,]	0	1	NA	0	0	2	1	1	2	NA	2
##	[42,]	NA	0	0	NA	0	0	1	NA	NA	0	0
##	[43,]	0	1	2	NA	NA	2	0	2	NA	1	0
##	[44,]	2	1	NA	NA	NA	0	2	1	0	1	0
##	[45,]	0	2	NA	NA	NA	NA	NA	0	0	NA	0
##	[46,]	2	NA	NA	0	2	0	1	2	1	2	2
##	[47,]	0	1	1	2	NA	2	2	2	0	NA	1
##	[48,]	0	0	0	NA	0	NA	0	2	1	0	0
##	[49,]	NA	NA	NA	0	NA	0	0	0	1	1	NA
##	[50,]	NA	NA	0	NA	0	0	NA	0	0	0	1
##		[,36]	[,37]	[,38]	[,39]	[,40]	[,41]	[,42]	[,43]	[,44]	[,45]	[,46]
##	[1,]	0	NA	0	NA	2	NA	NA	0	NA	NA	0
##	[2,]	0	1	NA	0	NA	0	0	1	NA	1	1
##	[3,]	2	NA	2	NA	0	2	2	NA	1	NA	1
##	[4,]	0	2	1	NA	0	1	2	NA	2	NA	NA
##	[5,]	1	2	2	1	0	2	0	1	0	NA	NA
##	[6,]	0	2	0	0	2	1	0	0	2	2	NA
##	[7,]	1	0	1	2	2	2	2	0	NA	1	2
##	[8,]	0	0	NA	1	NA	1	0	1	NA	0	NA
##	[9,]	0	0	NA	1	2	0	0	1	0	0	NA
##	[10,]	1	0	2	NA	1	1	0	NA	2	0	0
##	[11,]	NA	2	2	NA	1	1	2	NA	1	NA	NA
##	[12,]	0	NA	2	NA	NA	2	NA	0	0	0	2
##	[13,]	0	2	NA	NA	2	0	NA	NA	NA	NA	NA
##	[14,]	NA	2	2	NA	2	NA	0	0	1	0	NA
##	[15,]	2	NA	0	NA	NA	0	NA	NA	NA	NA	0
##	[16,]	NA	NA	0	0	0	0	NA	NA	NA	0	1
##	[17,]	0	1	0	2	1	0	1	1	NA	2	NA
##	[18,]	2	1	0	0	0	1	2	1	2	NA	0
##	[19,]	NA	NA	NA	NA	NA	0	NA	NA	NA	NA	NA
##	[20,]	0	2	2	NA	1	2	NA	2	NA	NA	NA
##	[21,]	0	NA	NA	0	1	0	1	NA	1	0	0
##	[22,]	NA	0	0	2	2	NA	2	2	NA	2	0
##	[23,]	NA	1	NA	2	NA	NA	1	NA	NA	0	1
##	[24,]	1	2	0	1	0	0	2	1	0	NA	2
##	[25,]	0	1	NA	0	1	0	NA	1	NA	0	NA
##	[26,]	0	0	NA	1	0	0	NA	0	0	0	NA
##	[27,]	2	NA	NA	1	1	NA	1	2	1	0	0
##	[28,]	0	2	NA	NA	NA	0	NA	2	1	0	NA
##	[29,]	1	1	0	2	1	1	0	1	NA	2	2
##	[30,]	2	NA	NA	0	NA	2	1	1	NA	NA	2
##	[31,]	1	0	0	NA	0	NA	2	0	2	1	0
##	[32,]	0	NA	2	1	0	2	NA	1	0	0	2
##	[33,]	NA	NA	NA	0	0	2	NA	1	2	NA	NA
##	[34,]	0	NA	2	NA	1	1	NA	NA	NA	NA	1
##	[35,]	NA	2	NA	NA	1	0	NA	2	NA	NA	0
##	[36,]	0	2	0	2	NA	0	1	NA	NA	0	NA
##	[37,]	NA	NA	0	2	0	NA	2	2	0	NA	NA
##	[38,]	2	NA	NA	NA	0	2	0	1	0	0	2
##	[39,]	0	NA	0	NA	0	2	0	0	2	0	2
##	[40,]	1	NA	0	0	NA	1	0	1	0	1	1
##	[41,]	2	0	0	0	2	0	1	NA	1	0	0

##	[42,]	1	1	2	1	2	NA	0	NA	0	0	0
##	[43,]	1	NA	0	1	NA	2	0	0	1	NA	1
##	[44,]	NA	NA	0	2	0	0	0	0	0	NA	NA
##	[45,]	1	0	NA	1	NA	NA	0	2	NA	0	NA
##	[46,]	0	1	2	0	0	0	NA	0	0	2	NA
##	[47,]	NA	0	1	0	0	2	NA	1	0	1	2
##	[48,]	1	1	1	2	0	NA	1	NA	1	1	1
##	[49,]	NA	0	NA	0	2	NA	1	0	0	1	NA
##	[50,]	NA	0	2	1	0	NA	0	1	0	0	0
##	[,47] [,48] [,49] [,50]											
##	[1,]	NA	NA	NA	2							
##	[2,]	1	0	2	0							
##	[3,]	1	NA	NA	NA							
##	[4,]	2	1	NA	NA							
##	[5,]	0	NA	NA	2							
##	[6,]	NA	2	0	NA							
##	[7,]	NA	1	2	2							
##	[8,]	NA	1	0	NA							
##	[9,]	0	1	2	1							
##	[10,]	0	1	0	NA							
##	[11,]	NA	NA	2	NA							
##	[12,]	0	NA	NA	NA							
##	[13,]	0	2	2	NA							
##	[14,]	0	NA	NA	1							
##	[15,]	0	0	1	1							
##	[16,]	1	2	0	1							
##	[17,]	0	0	1	1							
##	[18,]	1	NA	0	2							
##	[19,]	NA	NA	0	1							
##	[20,]	NA	NA	NA	1							
##	[21,]	0	1	0	2							
##	[22,]	0	0	0	0							
##	[23,]	NA	0	0	0							
##	[24,]	NA	0	2	NA							
##	[25,]	0	0	1	2							
##	[26,]	1	0	2	0							
##	[27,]	1	2	0	NA							
##	[28,]	NA	0	NA	NA							
##	[29,]	2	0	NA	NA							
##	[30,]	2	NA	0	1							
##	[31,]	NA	NA	NA	1							
##	[32,]	NA	2	NA	2							
##	[33,]	0	2	2	0							
##	[34,]	1	NA	0	NA							
##	[35,]	0	2	NA	NA							
##	[36,]	NA	0	1	0							
##	[37,]	1	NA	NA	2							
##	[38,]	2	1	1	1							
##	[39,]	2	NA	2	NA							
##	[40,]	1	NA	NA	NA							
##	[41,]	0	1	2	0							
##	[42,]	NA	NA	1	0							
##	[43,]	2	NA	1	2							
##	[44,]	2	1	0	2							

```
## [45,]      2    NA    NA      1
## [46,]     NA     0     0      0
## [47,]      0     0    NA      2
## [48,]     NA    NA    NA      0
## [49,]     NA     2    NA     NA
## [50,]      0    NA     2     NA
```

```
table(R)
```

```
## R
##      0      1      2
## 847 439 457
```

Use the `testthat` library to test that this worked correctly by ensuring the number of missing entries is between the 0.5%ile and 99.5%ile of the appropriate binomial.

```
NA_Count = 0
```

```
#Counts the # of NA's in R, stores it in NA_Count
```

```
for(i in R){
  if(is.na(i)){
    NA_Count = NA_Count + 1
  }
}
```

```
if(expect_lt(NA_Count, qbinom(.995, n*n, 0.3)) & expect_gt(NA_Count, qbinom(.005, n*n, 0.3))){
  cat("The number of missing entries is between the 0.5%ile and 99.5%ile of the appropriate binomial.")
} else{
  cat("The number of missing entries is NOT between the 0.5%ile and 99.5%ile of the appropriate binomial.")
}
```

The number of missing entries is between the 0.5%ile and 99.5%ile of the appropriate binomial.

- Sort the rows matrix R by the largest row sum to lowest. Be careful about the NA's!

```
R
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]   NA     0     2     0     0     0     0   NA   NA     NA     0     0    NA
## [2,]     1    NA     0     0    NA     0    NA     0   NA     0     1     1     2
## [3,]     1    NA     0     2     1     0     0     0     2     0     0     0     0
## [4,]   NA     0     2    NA    NA     1     1     0     1     0    NA     2     0
## [5,]     2     0     2     0     2    NA     1    NA     2    NA     1     0     1
## [6,]     0    NA     0     0    NA     1    NA     0     1    NA     1     1     2
## [7,]   NA     2    NA     2     0     0     1     0    NA     2    NA     1     0
## [8,]     2    NA     2    NA     1     0     2     1     1     1     0     2     2
## [9,]     0     1    NA    NA    NA    NA     2    NA     1     2     0     2    NA
## [10,]    1     0     2    NA     1     2     2    NA    NA     0     1     2    NA
## [11,]    0    NA     0     0     0     0     0     0    NA    NA     0     0     0
## [12,]    0     0    NA    NA     1     1     1     2    NA     1     1    NA     0
## [13,]   NA     0     2     0    NA     1     0    NA     2    NA     0     0     0
## [14,]     1     0     0    NA     2     1     0    NA     1     2     2    NA     0
## [15,]   NA     1     2    NA     1     2    NA    NA     1     1     2     2     2
## [16,]     2     1     2     2     1     1    NA     2     2     0     0    NA     0
## [17,]     2     2     1     1     0     0     0     0     0     0     0     2     1
## [18,]     1     0     2     1     0     0     0     0     1    NA     2    NA     0
## [19,]     0     2     1     2     0    NA     0     2    NA     0     1    NA     0
## [20,]     0     0     0     0     2     0     2     0    NA    NA     1     2     0
```

## [21,]	0	0	1	0	0	NA	1	0	0	1	0	0	2
## [22,]	0	NA	2	0	NA	1	NA	NA	NA	NA	NA	0	2
## [23,]	NA	2	1	0	0	NA	NA	NA	0	2	1	2	0
## [24,]	0	2	1	0	NA	NA	0	0	1	2	0	NA	NA
## [25,]	NA	0	0	NA	2	0	1	1	1	0	NA	2	NA
## [26,]	2	0	1	NA	0	2	0	NA	2	0	2	0	NA
## [27,]	NA	0	NA	1	0	NA	0	NA	NA	1	0	0	2
## [28,]	1	1	1	NA	2	0	2	NA	NA	0	2	NA	2
## [29,]	1	0	NA	2	NA	2	0	0	0	2	1	1	1
## [30,]	NA	0	0	1	1	1	NA	0	NA	0	0	2	NA
## [31,]	NA	0	NA	1	0	2	2	1	NA	2	2	0	0
## [32,]	0	NA	2	1	NA	0	0	NA	2	0	NA	NA	0
## [33,]	2	0	NA	0	0	NA	1	0	NA	2	2	NA	2
## [34,]	NA	1	2	2	0	0	1	0	1	NA	0	0	NA
## [35,]	2	1	0	NA	NA	2	0	1	NA	NA	1	2	0
## [36,]	NA	0	2	NA	1	2	0	2	0	1	NA	NA	1
## [37,]	2	NA	2	NA	0	2	0	NA	NA	0	0	0	NA
## [38,]	0	NA	1	0	NA	0	NA	2	0	0	0	0	NA
## [39,]	2	2	1	2	2	1	2	1	2	1	2	0	0
## [40,]	NA	NA	1	NA	1	2	0	0	1	0	NA	0	0
## [41,]	NA	NA	NA	1	1	2	2	NA	2	NA	0	NA	1
## [42,]	0	1	NA	NA	NA	0	NA	0	0	0	1	NA	0
## [43,]	2	0	0	2	NA	NA	0	NA	1	NA	1	1	NA
## [44,]	2	0	0	1	0	1	0	NA	1	NA	1	2	2
## [45,]	1	2	1	0	0	0	NA	0	NA	0	NA	0	NA
## [46,]	NA	NA	2	NA	1	0	NA	0	NA	0	NA	NA	NA
## [47,]	1	1	NA	NA	NA	2	2	2	0	2	2	NA	NA
## [48,]	NA	0	NA	1	NA	NA	0	NA	NA	0	0	NA	1
## [49,]	NA	0	NA	NA	0	NA	NA	0	0	2	NA	2	1
## [50,]	0	0	0	0	NA	1	0	0	0	1	2	2	1
##	[,14]	[,15]	[,16]	[,17]	[,18]	[,19]	[,20]	[,21]	[,22]	[,23]	[,24]		
## [1,]	NA	1	2	0	1	1	0	2	2	0	1		
## [2,]	NA	NA	1	NA	1	NA	NA	2	NA	0	NA		
## [3,]	0	1	0	0	1	0	NA	2	1	0	0		
## [4,]	0	1	0	2	0	0	NA	0	0	NA	2		
## [5,]	1	0	1	NA	1	2	2	1	1	NA	NA		
## [6,]	0	NA	0	1	2	NA	NA	0	1	0	0		
## [7,]	NA	0	NA	NA	0	NA	NA	2	1	2	NA		
## [8,]	NA	NA	0	0	NA	NA	2	NA	1	NA	0		
## [9,]	1	0	1	1	0	1	NA	NA	NA	0	2		
## [10,]	NA	0	0	1	NA	0	1	2	0	NA	0		
## [11,]	2	2	1	0	0	0	0	NA	1	2	0		
## [12,]	0	NA	0	2	2	1	NA	0	2	2	2		
## [13,]	2	0	0	NA	2	NA	0	NA	2	1	0		
## [14,]	2	1	2	2	2	0	2	0	NA	1	1		
## [15,]	2	1	0	0	1	NA	0	NA	2	2	1		
## [16,]	NA	0	1	NA	2	0	0	NA	2	NA	2		
## [17,]	NA	0	0	0	0	0	1	2	0	0	NA		
## [18,]	0	NA	2	1	NA	2	0	NA	0	0	2		
## [19,]	0	2	NA	0	0	NA	2	0	1	0	1		
## [20,]	0	0	NA	NA	1	0	0	NA	1	0	0		
## [21,]	NA	NA	NA	0	NA	NA	NA	0	NA	NA	NA		
## [22,]	2	NA	2	1	NA	1	NA	NA	0	0	0		
## [23,]	2	NA	2	2	NA	2	NA	0	0	2	2		

## [24,]	0	NA	0	2	0	NA	2	NA	NA	0	0
## [25,]	NA	1	0	NA	NA	0	0	0	0	0	0
## [26,]	2	0	0	0	1	0	NA	1	NA	0	0
## [27,]	NA	2	0	NA	NA	NA	2	0	1	2	NA
## [28,]	2	0	NA	1	0	NA	0	0	0	NA	0
## [29,]	0	NA	0	2	2	2	NA	NA	NA	0	0
## [30,]	NA	NA	NA	2	1	0	2	NA	0	NA	0
## [31,]	1	NA	0	0	NA	2	NA	NA	1	NA	2
## [32,]	0	2	0	NA	NA	0	1	NA	NA	0	2
## [33,]	0	2	0	NA	NA	0	2	NA	0	0	0
## [34,]	0	0	0	NA	0	2	0	2	0	1	2
## [35,]	0	0	2	2	0	2	0	0	NA	1	1
## [36,]	0	0	NA	0	1	2	1	1	NA	0	2
## [37,]	2	2	0	0	0	NA	NA	0	2	NA	0
## [38,]	0	2	2	0	1	0	NA	NA	NA	1	2
## [39,]	NA	1	1	0	NA	0	NA	0	2	NA	NA
## [40,]	2	1	NA	2	NA	1	NA	1	0	NA	0
## [41,]	0	0	NA	1	2	0	0	2	NA	NA	0
## [42,]	1	2	0	1	NA	0	NA	NA	NA	NA	NA
## [43,]	0	0	0	0	0	NA	0	0	NA	0	NA
## [44,]	1	1	1	0	2	0	NA	2	0	0	NA
## [45,]	0	0	0	2	2	NA	0	0	NA	2	0
## [46,]	0	NA	NA	0	NA	1	0	2	2	2	0
## [47,]	NA	1	NA	2	0	NA	NA	NA	2	0	NA
## [48,]	NA	2	1	0	NA	NA	0	0	0	1	2
## [49,]	0	NA	0	NA	0	NA	0	0	0	NA	0
## [50,]	0	2	2	NA	1	0	NA	0	NA	1	0
##	[,25]	[,26]	[,27]	[,28]	[,29]	[,30]	[,31]	[,32]	[,33]	[,34]	[,35]
## [1,]	0	0	NA	0	NA	0	0	NA	0	2	0
## [2,]	2	NA	1	NA	1	0	0	0	1	2	0
## [3,]	2	0	NA	0	NA	0	NA	1	NA	2	0
## [4,]	1	NA	0	2	0	0	NA	0	NA	NA	0
## [5,]	1	2	NA	0	2	0	2	2	0	NA	NA
## [6,]	NA	0	1	0	NA	0	0	2	2	1	2
## [7,]	0	0	NA	NA	2	NA	0	0	0	NA	0
## [8,]	NA	NA	0	NA	1	2	NA	1	0	2	0
## [9,]	0	0	0	NA	0	2	0	2	1	0	NA
## [10,]	1	1	0	2	0	1	0	1	NA	0	2
## [11,]	1	0	2	NA	0	NA	1	0	1	NA	0
## [12,]	0	NA	1	0	0	0	1	NA	2	0	NA
## [13,]	1	0	1	1	1	1	NA	1	1	1	2
## [14,]	NA	1	NA	0	NA	1	NA	2	0	2	NA
## [15,]	2	NA	NA	2	2	2	1	0	NA	0	2
## [16,]	2	NA	NA	1	2	NA	1	0	NA	2	0
## [17,]	NA	0	2	NA	0	NA	1	NA	NA	0	1
## [18,]	0	NA	2	NA	2	0	0	0	NA	0	0
## [19,]	1	0	2	1	1	1	1	1	0	NA	NA
## [20,]	NA	0	0	0	0	0	NA	NA	0	NA	1
## [21,]	2	0	2	NA	NA	0	0	0	0	NA	1
## [22,]	0	0	0	0	1	0	1	0	0	NA	2
## [23,]	0	0	0	2	1	NA	1	0	1	0	1
## [24,]	0	1	0	NA	0	0	2	0	NA	0	NA
## [25,]	0	0	1	2	0	1	NA	0	1	0	1
## [26,]	1	0	NA	NA	2	0	0	NA	1	2	1

## [27,]	0	2	0	NA	0	0	0	0	NA	2	0
## [28,]	2	1	0	1	NA	1	0	1	0	0	2
## [29,]	NA	0	NA	NA	1	1	0	NA	1	2	0
## [30,]	2	NA	0	0	2	2	0	NA	1	NA	NA
## [31,]	NA	0	0	NA	0	1	1	1	NA	2	NA
## [32,]	0	NA	1	2	NA	0	0	NA	2	NA	2
## [33,]	1	0	NA	0	1	0	NA	0	NA	NA	NA
## [34,]	2	NA	2	1	1	1	0	1	2	0	0
## [35,]	NA	0	0	0	2	0	0	2	1	0	NA
## [36,]	NA	0	2	1	1	0	0	0	NA	1	NA
## [37,]	NA	2	NA	NA	NA	NA	2	2	0	NA	0
## [38,]	0	2	1	NA	NA	NA	0	NA	0	NA	1
## [39,]	NA	1	2	0	2	2	NA	0	2	0	NA
## [40,]	NA	0	1	2	2	NA	1	0	2	1	0
## [41,]	0	1	NA	0	0	2	1	1	2	NA	2
## [42,]	NA	0	0	NA	0	0	1	NA	NA	0	0
## [43,]	0	1	2	NA	NA	2	0	2	NA	1	0
## [44,]	2	1	NA	NA	NA	0	2	1	0	1	0
## [45,]	0	2	NA	NA	NA	NA	NA	0	0	NA	0
## [46,]	2	NA	NA	0	2	0	1	2	1	2	2
## [47,]	0	1	1	2	NA	2	2	2	0	NA	1
## [48,]	0	0	0	NA	0	NA	0	2	1	0	0
## [49,]	NA	NA	NA	0	NA	0	0	0	1	1	NA
## [50,]	NA	NA	0	NA	0	0	NA	0	0	0	1
##	[,36]	[,37]	[,38]	[,39]	[,40]	[,41]	[,42]	[,43]	[,44]	[,45]	[,46]
## [1,]	0	NA	0	NA	2	NA	NA	0	NA	NA	0
## [2,]	0	1	NA	0	NA	0	0	1	NA	1	1
## [3,]	2	NA	2	NA	0	2	2	NA	1	NA	1
## [4,]	0	2	1	NA	0	1	2	NA	2	NA	NA
## [5,]	1	2	2	1	0	2	0	1	0	NA	NA
## [6,]	0	2	0	0	2	1	0	0	2	2	NA
## [7,]	1	0	1	2	2	2	2	0	NA	1	2
## [8,]	0	0	NA	1	NA	1	0	1	NA	0	NA
## [9,]	0	0	NA	1	2	0	0	1	0	0	NA
## [10,]	1	0	2	NA	1	1	0	NA	2	0	0
## [11,]	NA	2	2	NA	1	1	2	NA	1	NA	NA
## [12,]	0	NA	2	NA	NA	2	NA	0	0	0	2
## [13,]	0	2	NA	NA	2	0	NA	NA	NA	NA	NA
## [14,]	NA	2	2	NA	2	NA	0	0	1	0	NA
## [15,]	2	NA	0	NA	NA	0	NA	NA	NA	NA	0
## [16,]	NA	NA	0	0	0	0	NA	NA	NA	0	1
## [17,]	0	1	0	2	1	0	1	1	NA	2	NA
## [18,]	2	1	0	0	0	1	2	1	2	NA	0
## [19,]	NA	NA	NA	NA	NA	0	NA	NA	NA	NA	NA
## [20,]	0	2	2	NA	1	2	NA	2	NA	NA	NA
## [21,]	0	NA	NA	0	1	0	1	NA	1	0	0
## [22,]	NA	0	0	2	2	NA	2	2	NA	2	0
## [23,]	NA	1	NA	2	NA	NA	1	NA	NA	0	1
## [24,]	1	2	0	1	0	0	2	1	0	NA	2
## [25,]	0	1	NA	0	1	0	NA	1	NA	0	NA
## [26,]	0	0	NA	1	0	0	NA	0	0	0	NA
## [27,]	2	NA	NA	1	1	NA	1	2	1	0	0
## [28,]	0	2	NA	NA	NA	0	NA	2	1	0	NA
## [29,]	1	1	0	2	1	1	0	1	NA	2	2

##	[30,]	2	NA	NA	0	NA	2	1	1	NA	NA	2
##	[31,]	1	0	0	NA	0	NA	2	0	2	1	0
##	[32,]	0	NA	2	1	0	2	NA	1	0	0	2
##	[33,]	NA	NA	NA	0	0	2	NA	1	2	NA	NA
##	[34,]	0	NA	2	NA	1	1	NA	NA	NA	NA	1
##	[35,]	NA	2	NA	NA	1	0	NA	2	NA	NA	0
##	[36,]	0	2	0	2	NA	0	1	NA	NA	0	NA
##	[37,]	NA	NA	0	2	0	NA	2	2	0	NA	NA
##	[38,]	2	NA	NA	NA	0	2	0	1	0	0	2
##	[39,]	0	NA	0	NA	0	2	0	0	2	0	2
##	[40,]	1	NA	0	0	NA	1	0	1	0	1	1
##	[41,]	2	0	0	0	2	0	1	NA	1	0	0
##	[42,]	1	1	2	1	2	NA	0	NA	0	0	0
##	[43,]	1	NA	0	1	NA	2	0	0	1	NA	1
##	[44,]	NA	NA	0	2	0	0	0	0	0	NA	NA
##	[45,]	1	0	NA	1	NA	NA	0	2	NA	0	NA
##	[46,]	0	1	2	0	0	0	NA	0	0	2	NA
##	[47,]	NA	0	1	0	0	2	NA	1	0	1	2
##	[48,]	1	1	1	2	0	NA	1	NA	1	1	1
##	[49,]	NA	0	NA	0	2	NA	1	0	0	1	NA
##	[50,]	NA	0	2	1	0	NA	0	1	0	0	0
##	[,47] [,48] [,49] [,50]											
##	[1,]	NA	NA	NA	2							
##	[2,]	1	0	2	0							
##	[3,]	1	NA	NA	NA							
##	[4,]	2	1	NA	NA							
##	[5,]	0	NA	NA	2							
##	[6,]	NA	2	0	NA							
##	[7,]	NA	1	2	2							
##	[8,]	NA	1	0	NA							
##	[9,]	0	1	2	1							
##	[10,]	0	1	0	NA							
##	[11,]	NA	NA	2	NA							
##	[12,]	0	NA	NA	NA							
##	[13,]	0	2	2	NA							
##	[14,]	0	NA	NA	1							
##	[15,]	0	0	1	1							
##	[16,]	1	2	0	1							
##	[17,]	0	0	1	1							
##	[18,]	1	NA	0	2							
##	[19,]	NA	NA	0	1							
##	[20,]	NA	NA	NA	1							
##	[21,]	0	1	0	2							
##	[22,]	0	0	0	0							
##	[23,]	NA	0	0	0							
##	[24,]	NA	0	2	NA							
##	[25,]	0	0	1	2							
##	[26,]	1	0	2	0							
##	[27,]	1	2	0	NA							
##	[28,]	NA	0	NA	NA							
##	[29,]	2	0	NA	NA							
##	[30,]	2	NA	0	1							
##	[31,]	NA	NA	NA	1							
##	[32,]	NA	2	NA	2							

```
## [33,]    0    2    2    0
## [34,]    1   NA    0   NA
## [35,]    0    2   NA   NA
## [36,]   NA    0    1    0
## [37,]    1   NA   NA    2
## [38,]    2    1    1    1
## [39,]    2   NA    2   NA
## [40,]    1   NA   NA   NA
## [41,]    0    1    2    0
## [42,]   NA   NA    1    0
## [43,]    2   NA    1    2
## [44,]    2    1    0    2
## [45,]    2   NA   NA    1
## [46,]   NA    0    0    0
## [47,]    0    0   NA    2
## [48,]   NA   NA   NA    0
## [49,]   NA    2   NA   NA
## [50,]    0   NA    2   NA
```

```
#Creates empty vector to store row sums
```

```
row_Sums = rep(NA, nrow(R))
```

```
#Stores all 50 row sums in vector row_Sums
```

```
for (i in (1:nrow(R))){
  row_Sums[i] = sum(R[i,], na.rm = TRUE)
}
```

```
#Assigns the row sums as the names of the rows
```

```
row.names(R) = row_Sums
```

```
#Orders the row names by decreasing value, and creates new matrix
```

```
R = R[order(rownames(R), decreasing = TRUE),]
```

```
R
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## 41     2     2     1     2     2     1     2     1     2     1     2     0     0
## 40     2     0     2     0     2    NA     1    NA     2    NA     1     0     1
## 38    NA     1     2    NA     1     2    NA    NA     1     1     2     2     2
## 37     1     1    NA    NA    NA     2     2     2     0     2     2    NA    NA
## 36     1     0     0    NA     2     1     0    NA     1     2     2    NA     0
## 34     1     0    NA     2    NA     2     0     0     0     2     1     1     1
## 33    NA     2    NA     2     0     0     1     0    NA     2    NA     1     0
## 33     2     1     2     2     1     1    NA     2     2     0     0    NA     0
## 32    NA    NA    NA     1     1     2     2    NA     2    NA     0    NA     1
## 31     1     0     2    NA     1     2     2    NA    NA     0     1     2    NA
## 31    NA     2     1     0     0    NA    NA    NA     0     2     1     2     0
## 31     2     0     0     1     0     1     0    NA     1    NA     1     2     2
## 30    NA     0     2     0    NA     1     0    NA     2    NA     0     0     0
## 30     1     0     2     1     0     0     0     0     1    NA     2    NA     0
## 30    NA     1     2     2     0     0     1     0     1    NA     0     0    NA
## 29     0    NA     0     0    NA     1    NA     0     1    NA     1     1     2
## 29     0    NA     2     1    NA     0     0    NA     2     0    NA    NA     0
## 29     2     1     0    NA    NA     2     0     1    NA    NA     1     2     0
## 28     0     0    NA    NA     1     1     1     2    NA     1     1    NA     0
```

## 28	NA	0	0	1	1	1	NA	0	NA	0	0	2	NA
## 28	NA	0	NA	1	0	2	2	1	NA	2	2	0	0
## 27	1	NA	0	2	1	0	0	0	2	0	0	0	0
## 27	2	NA	2	NA	1	0	2	1	1	1	0	2	2
## 27	0	1	NA	NA	NA	NA	2	NA	1	2	0	2	NA
## 27	1	1	1	NA	2	0	2	NA	NA	0	2	NA	2
## 27	NA	0	2	NA	1	2	0	2	0	1	NA	NA	1
## 27	2	NA	2	NA	0	2	0	NA	NA	0	0	0	NA
## 27	0	NA	1	0	NA	0	NA	2	0	0	0	0	NA
## 27	NA	NA	1	NA	1	2	0	0	1	0	NA	0	0
## 27	NA	NA	2	NA	1	0	NA	0	NA	0	NA	NA	NA
## 26	NA	0	2	NA	NA	1	1	0	1	0	NA	2	0
## 26	2	2	1	1	0	0	0	0	0	0	0	2	1
## 26	NA	0	NA	1	0	NA	0	NA	NA	1	0	0	2
## 26	2	0	0	2	NA	NA	0	NA	1	NA	1	1	NA
## 25	0	NA	2	0	NA	1	NA	NA	NA	NA	NA	0	2
## 24	0	NA	0	0	0	0	0	0	NA	NA	0	0	0
## 24	0	2	1	0	NA	NA	0	0	1	2	0	NA	NA
## 24	2	0	1	NA	0	2	0	NA	2	0	2	0	NA
## 24	2	0	NA	0	0	NA	1	0	NA	2	2	NA	2
## 23	1	NA	0	0	NA	0	NA	0	NA	0	1	1	2
## 23	0	2	1	2	0	NA	0	2	NA	0	1	NA	0
## 20	0	0	0	0	2	0	2	0	NA	NA	1	2	0
## 20	NA	0	0	NA	2	0	1	1	1	0	NA	2	NA
## 20	NA	0	NA	1	NA	NA	0	NA	NA	0	0	NA	1
## 20	0	0	0	0	NA	1	0	0	0	1	2	2	1
## 19	1	2	1	0	0	0	NA	0	NA	0	NA	0	NA
## 18	NA	0	2	0	0	0	0	NA	NA	NA	0	0	NA
## 16	0	0	1	0	0	NA	1	0	0	1	0	0	2
## 15	0	1	NA	NA	NA	0	NA	0	0	0	1	NA	0
## 13	NA	0	NA	NA	0	NA	NA	0	0	2	NA	2	1
##	[,14]	[,15]	[,16]	[,17]	[,18]	[,19]	[,20]	[,21]	[,22]	[,23]	[,24]	[,25]	
## 41	NA	1	1	0	NA	0	NA	0	2	NA	NA	NA	
## 40	1	0	1	NA	1	2	2	1	1	NA	NA	1	
## 38	2	1	0	0	1	NA	0	NA	2	2	1	2	
## 37	NA	1	NA	2	0	NA	NA	NA	2	0	NA	0	
## 36	2	1	2	2	2	0	2	0	NA	1	1	NA	
## 34	0	NA	0	2	2	2	NA	NA	NA	0	0	NA	
## 33	NA	0	NA	NA	0	NA	NA	2	1	2	NA	0	
## 33	NA	0	1	NA	2	0	0	NA	2	NA	2	2	
## 32	0	0	NA	1	2	0	0	2	NA	NA	0	0	
## 31	NA	0	0	1	NA	0	1	2	0	NA	0	1	
## 31	2	NA	2	2	NA	2	NA	0	0	2	2	0	
## 31	1	1	1	0	2	0	NA	2	0	0	NA	2	
## 30	2	0	0	NA	2	NA	0	NA	2	1	0	1	
## 30	0	NA	2	1	NA	2	0	NA	0	0	2	0	
## 30	0	0	0	NA	0	2	0	2	0	1	2	2	
## 29	0	NA	0	1	2	NA	NA	0	1	0	0	NA	
## 29	0	2	0	NA	NA	0	1	NA	NA	0	2	0	
## 29	0	0	2	2	0	2	0	0	NA	1	1	NA	
## 28	0	NA	0	2	2	1	NA	0	2	2	2	0	
## 28	NA	NA	NA	2	1	0	2	NA	0	NA	0	2	
## 28	1	NA	0	0	NA	2	NA	NA	1	NA	2	NA	
## 27	0	1	0	0	1	0	NA	2	1	0	0	2	

## 27	NA	NA	0	0	NA	NA	2	NA	1	NA	0	NA
## 27	1	0	1	1	0	1	NA	NA	NA	0	2	0
## 27	2	0	NA	1	0	NA	0	0	0	NA	0	2
## 27	0	0	NA	0	1	2	1	1	NA	0	2	NA
## 27	2	2	0	0	0	NA	NA	0	2	NA	0	NA
## 27	0	2	2	0	1	0	NA	NA	NA	1	2	0
## 27	2	1	NA	2	NA	1	NA	1	0	NA	0	NA
## 27	0	NA	NA	0	NA	1	0	2	2	2	0	2
## 26	0	1	0	2	0	0	NA	0	0	NA	2	1
## 26	NA	0	0	0	0	0	1	2	0	0	NA	NA
## 26	NA	2	0	NA	NA	NA	2	0	1	2	NA	0
## 26	0	0	0	0	0	NA	0	0	NA	0	NA	0
## 25	2	NA	2	1	NA	1	NA	NA	0	0	0	0
## 24	2	2	1	0	0	0	0	NA	1	2	0	1
## 24	0	NA	0	2	0	NA	2	NA	NA	0	0	0
## 24	2	0	0	0	1	0	NA	1	NA	0	0	1
## 24	0	2	0	NA	NA	0	2	NA	0	0	0	1
## 23	NA	NA	1	NA	1	NA	NA	2	NA	0	NA	2
## 23	0	2	NA	0	0	NA	2	0	1	0	1	1
## 20	0	0	NA	NA	1	0	0	NA	1	0	0	NA
## 20	NA	1	0	NA	NA	0	0	0	0	0	0	0
## 20	NA	2	1	0	NA	NA	0	0	0	1	2	0
## 20	0	2	2	NA	1	0	NA	0	NA	1	0	NA
## 19	0	0	0	2	2	NA	0	0	NA	2	0	0
## 18	NA	1	2	0	1	1	0	2	2	0	1	0
## 16	NA	NA	NA	0	NA	NA	NA	0	NA	NA	NA	2
## 15	1	2	0	1	NA	0	NA	NA	NA	NA	NA	NA
## 13	0	NA	0	NA	0	NA	0	0	0	NA	0	NA
##	[,26]	[,27]	[,28]	[,29]	[,30]	[,31]	[,32]	[,33]	[,34]	[,35]	[,36]	[,37]
## 41	1	2	0	2	2	NA	0	2	0	NA	0	NA
## 40	2	NA	0	2	0	2	2	0	NA	NA	1	2
## 38	NA	NA	2	2	2	1	0	NA	0	2	2	NA
## 37	1	1	2	NA	2	2	2	0	NA	1	NA	0
## 36	1	NA	0	NA	1	NA	2	0	2	NA	NA	2
## 34	0	NA	NA	1	1	0	NA	1	2	0	1	1
## 33	0	NA	NA	2	NA	0	0	0	NA	0	1	0
## 33	NA	NA	1	2	NA	1	0	NA	2	0	NA	NA
## 32	1	NA	0	0	2	1	1	2	NA	2	2	0
## 31	1	0	2	0	1	0	1	NA	0	2	1	0
## 31	0	0	2	1	NA	1	0	1	0	1	NA	1
## 31	1	NA	NA	NA	0	2	1	0	1	0	NA	NA
## 30	0	1	1	1	1	NA	1	1	1	2	0	2
## 30	NA	2	NA	2	0	0	0	NA	0	0	2	1
## 30	NA	2	1	1	1	0	1	2	0	0	0	NA
## 29	0	1	0	NA	0	0	2	2	1	2	0	2
## 29	NA	1	2	NA	0	0	NA	2	NA	2	0	NA
## 29	0	0	0	2	0	0	2	1	0	NA	NA	2
## 28	NA	1	0	0	0	1	NA	2	0	NA	0	NA
## 28	NA	0	0	2	2	0	NA	1	NA	NA	2	NA
## 28	0	0	NA	0	1	1	1	NA	2	NA	1	0
## 27	0	NA	0	NA	0	NA	1	NA	2	0	2	NA
## 27	NA	0	NA	1	2	NA	1	0	2	0	0	0
## 27	0	0	NA	0	2	0	2	1	0	NA	0	0
## 27	1	0	1	NA	1	0	1	0	0	2	0	2

## 27	0	2	1	1	0	0	0	NA	1	NA	0	2
## 27	2	NA	NA	NA	NA	2	2	0	NA	0	NA	NA
## 27	2	1	NA	NA	NA	0	NA	0	NA	1	2	NA
## 27	0	1	2	2	NA	1	0	2	1	0	1	NA
## 27	NA	NA	0	2	0	1	2	1	2	2	0	1
## 26	NA	0	2	0	0	NA	0	NA	NA	0	0	2
## 26	0	2	NA	0	NA	1	NA	NA	0	1	0	1
## 26	2	0	NA	0	0	0	0	NA	2	0	2	NA
## 26	1	2	NA	NA	2	0	2	NA	1	0	1	NA
## 25	0	0	0	1	0	1	0	0	NA	2	NA	0
## 24	0	2	NA	0	NA	1	0	1	NA	0	NA	2
## 24	1	0	NA	0	0	2	0	NA	0	NA	1	2
## 24	0	NA	NA	2	0	0	NA	1	2	1	0	0
## 24	0	NA	0	1	0	NA	0	NA	NA	NA	NA	NA
## 23	NA	1	NA	1	0	0	0	1	2	0	0	1
## 23	0	2	1	1	1	1	1	0	NA	NA	NA	NA
## 20	0	0	0	0	0	NA	NA	0	NA	1	0	2
## 20	0	1	2	0	1	NA	0	1	0	1	0	1
## 20	0	0	NA	0	NA	0	2	1	0	0	1	1
## 20	NA	0	NA	0	0	NA	0	0	0	1	NA	0
## 19	2	NA	NA	NA	NA	NA	0	0	NA	0	1	0
## 18	0	NA	0	NA	0	0	NA	0	2	0	0	NA
## 16	0	2	NA	NA	0	0	0	0	NA	1	0	NA
## 15	0	0	NA	0	0	1	NA	NA	0	0	1	1
## 13	NA	NA	0	NA	0	0	0	1	1	NA	NA	0
##	[,38]	[,39]	[,40]	[,41]	[,42]	[,43]	[,44]	[,45]	[,46]	[,47]	[,48]	[,49]
## 41	0	NA	0	2	0	0	2	0	2	2	NA	2
## 40	2	1	0	2	0	1	0	NA	NA	0	NA	NA
## 38	0	NA	NA	0	NA	NA	NA	NA	0	0	0	1
## 37	1	0	0	2	NA	1	0	1	2	0	0	NA
## 36	2	NA	2	NA	0	0	1	0	NA	0	NA	NA
## 34	0	2	1	1	0	1	NA	2	2	2	0	NA
## 33	1	2	2	2	2	0	NA	1	2	NA	1	2
## 33	0	0	0	0	NA	NA	NA	0	1	1	2	0
## 32	0	0	2	0	1	NA	1	0	0	0	1	2
## 31	2	NA	1	1	0	NA	2	0	0	0	1	0
## 31	NA	2	NA	NA	1	NA	NA	0	1	NA	0	0
## 31	0	2	0	0	0	0	0	NA	NA	2	1	0
## 30	NA	NA	2	0	NA	NA	NA	NA	NA	0	2	2
## 30	0	0	0	1	2	1	2	NA	0	1	NA	0
## 30	2	NA	1	1	NA	NA	NA	NA	1	1	NA	0
## 29	0	0	2	1	0	0	2	2	NA	NA	2	0
## 29	2	1	0	2	NA	1	0	0	2	NA	2	NA
## 29	NA	NA	1	0	NA	2	NA	NA	0	0	2	NA
## 28	2	NA	NA	2	NA	0	0	0	2	0	NA	NA
## 28	NA	0	NA	2	1	1	NA	NA	2	2	NA	0
## 28	0	NA	0	NA	2	0	2	1	0	NA	NA	NA
## 27	2	NA	0	2	2	NA	1	NA	1	1	NA	NA
## 27	NA	1	NA	1	0	1	NA	0	NA	NA	1	0
## 27	NA	1	2	0	0	1	0	0	NA	0	1	2
## 27	NA	NA	NA	0	NA	2	1	0	NA	NA	0	NA
## 27	0	2	NA	0	1	NA	NA	0	NA	NA	0	1
## 27	0	2	0	NA	2	2	0	NA	NA	1	NA	NA
## 27	NA	NA	0	2	0	1	0	0	2	2	1	1

## 27	0	0	NA	1	0	1	0	1	1	1	NA	NA
## 27	2	0	0	0	NA	0	0	2	NA	NA	0	0
## 26	1	NA	0	1	2	NA	2	NA	NA	2	1	NA
## 26	0	2	1	0	1	1	NA	2	NA	0	0	1
## 26	NA	1	1	NA	1	2	1	0	0	1	2	0
## 26	0	1	NA	2	0	0	1	NA	1	2	NA	1
## 25	0	2	2	NA	2	2	NA	2	0	0	0	0
## 24	2	NA	1	1	2	NA	1	NA	NA	NA	NA	2
## 24	0	1	0	0	2	1	0	NA	2	NA	0	2
## 24	NA	1	0	0	NA	0	0	0	NA	1	0	2
## 24	NA	0	0	2	NA	1	2	NA	NA	0	2	2
## 23	NA	0	NA	0	0	1	NA	1	1	1	0	2
## 23	NA	NA	NA	0	NA	NA	NA	NA	NA	NA	NA	0
## 20	2	NA	1	2	NA	2	NA	NA	NA	NA	NA	NA
## 20	NA	0	1	0	NA	1	NA	0	NA	0	0	1
## 20	1	2	0	NA	1	NA	1	1	1	NA	NA	NA
## 20	2	1	0	NA	0	1	0	0	0	0	NA	2
## 19	NA	1	NA	NA	0	2	NA	0	NA	2	NA	NA
## 18	0	NA	2	NA	NA	0	NA	NA	0	NA	NA	NA
## 16	NA	0	1	0	1	NA	1	0	0	0	1	0
## 15	2	1	2	NA	0	NA	0	0	0	NA	NA	1
## 13	NA	0	2	NA	1	0	0	1	NA	NA	2	NA
##	[,50]											
## 41	NA											
## 40	2											
## 38	1											
## 37	2											
## 36	1											
## 34	NA											
## 33	2											
## 33	1											
## 32	0											
## 31	NA											
## 31	0											
## 31	2											
## 30	NA											
## 30	2											
## 30	NA											
## 29	NA											
## 29	2											
## 29	NA											
## 28	NA											
## 28	1											
## 28	1											
## 27	NA											
## 27	NA											
## 27	1											
## 27	NA											
## 27	0											
## 27	2											
## 27	1											
## 27	NA											
## 27	0											
## 26	NA											

```
## 26      1
## 26     NA
## 26      2
## 25      0
## 24     NA
## 24     NA
## 24      0
## 24      0
## 23      0
## 23      1
## 20      1
## 20      2
## 20      0
## 20     NA
## 19      1
## 18      2
## 16      2
## 15      0
## 13     NA
```

Test using the following code.

```
for (i in 2 : n){
  expect_gte(sum(R[i - 1, ], na.rm = TRUE), sum(R[i, ], na.rm = TRUE))
}
```

- We will now learn the `apply` function. This is a handy function that saves writing for loops which should be eschewed in R. Use the `apply` function to compute a vector whose entries are the standard deviation of each row. Use the `apply` function to compute a vector whose entries are the standard deviation of each column. Be careful about the NA's!

```
sd_of_rows = apply(R, 1, sd, na.rm = TRUE)
sd_of_columns = apply(R, 2, sd, na.rm = TRUE)
```

- Use the `apply` function to compute a vector whose entries are the count of entries that are 1 or 2 in each column. Try to do this in one line.

```
#Passes every column into the function, if a value is greater than zero, converts it to a 1, finds the
apply(R, 2, function(x) { sum(ifelse(x > 0, 1, 0), na.rm = TRUE) } )
```

```
## [1] 20 15 26 17 17 22 17 11 21 18 23 19 18 15 20 16 19 20 14 12 14 19 15
## [24] 17 18 14 18 14 21 17 18 20 20 18 18 18 20 16 20 20 21 18 23 16 12 17
## [47] 17 17 18 22
```

- Use the `split` function to create a list whose keys are the column number and values are the vector of the columns. Look at the last example in the documentation `?split`.

```
#Creates list w/R columns as keys by splitting the data by the columns of R.
list_R_Columns = split(R, col(R), drop=TRUE)
list_R_Columns
```

```
## $`1`
## [1]  2  2 NA  1  1  1 NA  2 NA  1 NA  2 NA  1 NA  0  0  2  0 NA NA  1  2
## [24]  0  1 NA  2  0 NA NA NA  2 NA  2  0  0  0  2  2  1  0  0 NA NA  0  1
## [47] NA  0  0 NA
##
## $`2`
## [1]  2  0  1  1  0  0  2  1 NA  0  2  0  0  0  1 NA NA  1  0  0  0 NA NA
```



```

## [24] 1 1 0 NA NA NA NA 0 2 0 0 NA NA 2 0 0 NA 2 0 0 0 0 2
## [47] 0 0 1 0
##
## $`3`
## [1] 1 2 2 NA 0 NA NA 2 NA 2 1 0 2 2 2 0 2 0 NA 0 NA 0 2
## [24] NA 1 2 2 1 1 2 2 1 NA 0 2 0 1 1 NA 0 1 0 0 NA 0 1
## [47] 2 1 NA NA
##
## $`4`
## [1] 2 0 NA NA NA 2 2 2 1 NA 0 1 0 1 2 0 1 NA NA 1 1 2 NA
## [24] NA NA NA NA 0 NA NA NA 1 1 2 0 0 0 NA 0 0 2 0 NA 1 0 0
## [47] 0 0 NA NA
##
## $`5`
## [1] 2 2 1 NA 2 NA 0 1 1 1 0 0 NA 0 0 NA NA NA 1 1 0 1 1
## [24] NA 2 1 0 NA 1 1 NA 0 0 NA NA 0 NA 0 0 NA 0 2 2 NA NA 0
## [47] 0 0 NA 0
##
## $`6`
## [1] 1 NA 2 2 1 2 0 1 2 2 NA 1 1 0 0 1 0 2 1 1 2 0 0
## [24] NA 0 2 2 0 2 0 1 0 NA NA 1 0 NA 2 NA 0 NA 0 0 NA 1 0
## [47] 0 NA 0 NA
##
## $`7`
## [1] 2 1 NA 2 0 0 1 NA 2 2 NA 0 0 0 1 NA 0 0 1 NA 2 0 2
## [24] 2 2 0 0 NA 0 NA 1 0 0 0 NA 0 0 0 1 NA 0 2 1 0 0 NA
## [47] 0 1 NA NA
##
## $`8`
## [1] 1 NA NA 2 NA 0 0 2 NA NA NA NA NA 0 0 0 NA 1 2 0 1 0 1
## [24] NA NA 2 NA 2 0 0 0 0 NA NA NA 0 0 NA 0 0 2 0 1 NA 0 0
## [47] NA 0 0 0
##
## $`9`
## [1] 2 2 1 0 1 0 NA 2 2 NA 0 1 2 1 1 1 2 NA NA NA NA 2 1
## [24] 1 NA 0 NA 0 1 NA 1 0 NA 1 NA NA 1 2 NA NA NA NA 1 NA 0 NA
## [47] NA 0 0 0
##
## $`10`
## [1] 1 NA 1 2 2 2 2 0 NA 0 2 NA NA NA NA NA 0 NA 1 0 2 0 1
## [24] 2 0 1 0 0 0 0 0 0 1 NA NA NA 2 0 2 0 0 NA 0 0 1 0
## [47] NA 1 0 2
##
## $`11`
## [1] 2 1 2 2 2 1 NA 0 0 1 1 1 0 2 0 1 NA 1 1 0 2 0 0
## [24] 0 2 NA 0 0 NA NA NA 0 0 1 NA 0 0 2 2 1 1 1 NA 0 2 NA
## [47] 0 0 1 NA
##
## $`12`
## [1] 0 0 2 NA NA 1 1 NA NA 2 2 2 0 NA 0 1 NA 2 NA 2 0 0 2
## [24] 2 NA NA 0 0 0 NA 2 2 0 1 0 0 NA 0 NA 1 NA 2 2 NA 2 0
## [47] 0 0 NA 2
##
## $`13`

```

```

## [1] 0 1 2 NA 0 1 0 0 1 NA 0 2 0 0 NA 2 0 0 0 NA 0 0 2
## [24] NA 2 1 NA NA 0 NA 0 1 2 NA 2 0 NA NA 2 2 0 0 NA 1 1 NA
## [47] NA 2 0 1
##
## $`14`
## [1] NA 1 2 NA 2 0 NA NA 0 NA 2 1 2 0 0 0 0 0 0 NA 1 0 NA
## [24] 1 2 0 2 0 2 0 0 NA NA 0 2 2 0 2 0 NA 0 0 NA NA 0 0
## [47] NA NA 1 0
##
## $`15`
## [1] 1 0 1 1 1 NA 0 0 0 0 NA 1 0 NA 0 NA 2 0 NA NA NA 1 NA
## [24] 0 0 0 2 2 1 NA 1 0 2 0 NA 2 NA 0 2 NA 2 0 1 2 2 0
## [47] 1 NA 2 NA
##
## $`16`
## [1] 1 1 0 NA 2 0 NA 1 NA 0 2 1 0 2 0 0 0 2 0 NA 0 0 0
## [24] 1 NA NA 0 2 NA NA 0 0 0 0 2 1 0 0 0 1 NA NA 0 1 2 0
## [47] 2 NA 0 0
##
## $`17`
## [1] 0 NA 0 2 2 2 NA NA 1 1 2 0 NA 1 NA 1 NA 2 2 2 0 0 0
## [24] 1 1 0 0 0 2 0 2 0 NA 0 1 0 2 0 NA NA 0 NA NA 0 NA 2
## [47] 0 0 1 NA
##
## $`18`
## [1] NA 1 1 0 2 2 0 2 2 NA NA 2 2 NA 0 2 NA 0 2 1 NA 1 NA
## [24] 0 0 1 0 1 NA NA 0 0 NA 0 NA 0 0 1 NA 1 0 1 NA NA 1 2
## [47] 1 NA NA 0
##
## $`19`
## [1] 0 2 NA NA 0 2 NA 0 0 0 2 0 NA 2 2 NA 0 2 1 0 2 0 NA
## [24] 1 NA 2 NA 0 1 1 0 0 NA NA 1 0 NA 0 0 NA NA 0 0 NA 0 NA
## [47] 1 NA 0 NA
##
## $`20`
## [1] NA 2 0 NA 2 NA NA 0 0 1 NA NA 0 0 0 NA 1 0 NA 2 NA NA 2
## [24] NA 0 1 NA NA NA 0 NA 1 2 0 NA 0 2 NA 2 NA 2 0 0 0 NA 0
## [47] 0 NA NA 0
##
## $`21`
## [1] 0 1 NA NA 0 NA 2 NA 2 2 0 2 NA NA 2 0 NA 0 0 NA NA 2 NA
## [24] NA 0 1 0 NA 1 2 0 2 0 0 NA NA NA 1 NA 2 0 NA 0 0 0 0
## [47] 2 0 NA 0
##
## $`22`
## [1] 2 1 2 2 NA NA 1 2 NA 0 0 0 2 0 0 1 NA NA 2 0 1 1 1
## [24] NA 0 NA 2 NA 0 2 0 0 1 NA 0 1 NA NA 0 NA 1 1 0 0 NA NA
## [47] 2 NA NA 0
##
## $`23`
## [1] NA NA 2 0 1 0 2 NA NA NA 2 0 1 0 1 0 0 1 2 NA NA 0 NA
## [24] 0 NA 0 NA 1 NA 2 NA 0 2 0 0 2 0 0 0 0 0 0 0 1 1 2
## [47] 0 NA NA NA
##
##

```

```

## $`24`
## [1] NA NA 1 NA 1 0 NA 2 0 0 2 NA 0 2 2 0 2 1 2 0 2 0 0
## [24] 2 0 2 0 2 0 0 2 NA NA NA 0 0 0 0 0 NA 1 0 0 2 0 0
## [47] 1 NA NA 0
##
## $`25`
## [1] NA 1 2 0 NA NA 0 2 0 1 0 2 1 0 2 NA 0 NA 0 2 NA 2 NA
## [24] 0 2 NA NA 0 NA 2 1 NA 0 0 0 1 0 1 1 2 1 NA 0 0 NA 0
## [47] 0 2 NA NA
##
## $`26`
## [1] 1 2 NA 1 1 0 0 NA 1 1 0 1 0 NA NA 0 NA 0 NA NA 0 0 NA
## [24] 0 1 0 2 2 0 NA NA 0 2 1 0 0 1 0 0 NA 0 0 0 0 NA 2
## [47] 0 0 0 NA
##
## $`27`
## [1] 2 NA NA 1 NA NA NA NA NA 0 0 NA 1 2 2 1 1 0 1 0 0 NA 0
## [24] 0 0 2 NA 1 1 NA 0 2 0 2 0 2 0 NA NA 1 2 0 1 0 0 NA
## [47] NA 2 0 NA
##
## $`28`
## [1] 0 0 2 2 0 NA NA 1 0 2 2 NA 1 NA 1 0 2 0 0 0 NA 0 NA
## [24] NA 1 1 NA NA 2 0 2 NA NA NA 0 NA NA NA 0 NA 1 0 2 NA NA NA
## [47] 0 NA NA 0
##
## $`29`
## [1] 2 2 2 NA NA 1 2 2 0 0 1 NA 1 2 1 NA NA 2 0 2 0 NA 1
## [24] 0 NA 1 NA NA 2 2 0 0 0 NA 1 0 0 2 1 1 1 0 0 0 0 NA
## [47] NA NA 0 NA
##
## $`30`
## [1] 2 0 2 2 1 1 NA NA 2 1 NA 0 1 0 1 0 0 0 0 2 1 0 2
## [24] 2 1 0 NA NA NA 0 0 NA 0 2 0 NA 0 0 0 0 1 0 1 NA 0 NA
## [47] 0 0 0 0
##
## $`31`
## [1] NA 2 1 2 NA 0 0 1 1 0 1 2 NA 0 0 0 0 0 1 0 1 NA NA
## [24] 0 0 0 2 0 1 1 NA 1 0 0 1 1 2 0 NA 0 1 NA NA 0 NA NA
## [47] 0 0 1 0
##
## $`32`
## [1] 0 2 0 2 2 NA 0 0 1 1 0 1 1 0 1 2 NA 2 NA NA 1 1 1
## [24] 2 1 0 2 NA 0 2 0 NA 0 2 0 0 0 NA 0 0 1 NA 0 2 0 0
## [47] NA 0 NA 0
##
## $`33`
## [1] 2 0 NA 0 0 1 0 NA 2 NA 1 0 1 NA 2 2 2 1 2 1 NA NA 0
## [24] 1 0 NA 0 0 2 1 NA NA NA NA 0 1 NA 1 NA 1 0 0 1 1 0 0
## [47] 0 0 NA 1
##
## $`34`
## [1] 0 NA 0 NA 2 2 NA 2 NA 0 0 1 1 0 0 1 NA 0 0 NA 2 2 2
## [24] 0 0 1 NA NA 1 2 NA 0 2 1 NA NA 0 2 NA 2 NA NA 0 0 0 NA
## [47] 2 NA 0 1

```

```

##
## $`35`
## [1] NA NA 2 1 NA 0 0 0 2 2 1 0 2 0 0 2 2 NA NA NA NA 0 0
## [24] NA 2 NA 0 1 0 2 0 1 0 0 2 0 NA 1 NA 0 NA 1 1 0 1 0
## [47] 0 1 0 NA
##
## $`36`
## [1] 0 1 2 NA NA 1 1 NA 2 1 NA NA 0 2 0 0 0 NA 0 2 1 2 0
## [24] 0 0 0 NA 2 1 0 0 0 2 1 NA NA 1 0 NA 0 NA 0 0 1 NA 1
## [47] 0 0 1 NA
##
## $`37`
## [1] NA 2 NA 0 2 1 0 NA 0 0 1 NA 2 1 NA 2 NA 2 NA NA 0 NA 0
## [24] 0 2 2 NA NA NA 1 2 1 NA NA 0 2 2 0 NA 1 NA 2 1 1 0 0
## [47] NA NA 1 0
##
## $`38`
## [1] 0 2 0 1 2 0 1 0 0 2 NA 0 NA 0 2 0 2 NA 2 NA 0 2 NA
## [24] NA NA 0 0 NA 0 2 1 0 NA 0 0 2 0 NA NA NA NA 2 NA 1 2 NA
## [47] 0 NA 2 NA
##
## $`39`
## [1] NA 1 NA 0 NA 2 2 0 0 NA 2 2 NA 0 NA 0 1 NA NA 0 NA NA 1
## [24] 1 NA 2 2 NA 0 0 NA 2 1 1 2 NA 1 1 0 0 NA NA 0 2 1 1
## [47] NA 0 1 0
##
## $`40`
## [1] 0 0 NA 0 2 1 2 0 2 1 NA 0 2 0 1 2 0 1 NA NA 0 0 NA
## [24] 2 NA NA 0 0 NA 0 0 1 1 NA 2 1 0 0 0 NA NA 1 1 0 0 NA
## [47] 2 1 2 2
##
## $`41`
## [1] 2 2 0 2 NA 1 2 0 0 1 NA 0 0 1 1 1 2 0 2 2 NA 2 1
## [24] 0 0 0 NA 2 1 0 1 0 NA 2 NA 1 0 0 2 0 0 2 0 NA NA NA
## [47] NA 0 NA NA
##
## $`42`
## [1] 0 0 NA NA 0 0 2 NA 1 0 1 0 NA 2 NA 0 NA NA NA 1 2 2 0
## [24] 0 NA 1 2 0 0 NA 2 1 1 0 2 2 2 NA NA 0 NA NA NA 1 0 0
## [47] NA 1 0 1
##
## $`43`
## [1] 0 1 NA 1 0 1 0 NA NA NA NA 0 NA 1 NA 0 1 2 0 1 0 NA 1
## [24] 1 2 NA 2 1 1 0 NA 1 2 0 2 NA 1 0 1 1 NA 2 1 NA 1 2
## [47] 0 NA NA 0
##
## $`44`
## [1] 2 0 NA 0 1 NA NA NA 1 2 NA 0 NA 2 NA 2 0 NA 0 NA 2 1 NA
## [24] 0 1 NA 0 0 0 0 2 NA 1 1 NA 1 0 0 2 NA NA NA NA 1 0 NA
## [47] NA 1 0 0
##
## $`45`
## [1] 0 NA NA 1 0 2 1 0 0 0 0 NA NA NA NA 2 0 NA 0 NA 1 NA 0
## [24] 0 0 0 NA 0 1 2 NA 2 0 NA 2 NA NA 0 NA 1 NA NA 0 1 0 0

```

```
## [47] NA 0 0 1
##
## $`46`
## [1] 2 NA 0 2 NA 2 2 1 0 0 1 NA NA 0 1 NA 2 0 2 2 0 1 NA
## [24] NA NA NA NA 2 1 NA NA NA 0 1 0 NA 2 NA NA 1 NA NA NA 1 0 NA
## [47] 0 0 0 NA
##
## $`47`
## [1] 2 0 0 0 0 2 NA 1 0 0 NA 2 0 1 1 NA NA 0 0 2 NA 1 NA
## [24] 0 NA NA 1 2 1 NA 2 0 1 2 0 NA NA 1 0 1 NA NA 0 NA 0 2
## [47] NA 0 NA NA
##
## $`48`
## [1] NA NA 0 0 NA 0 1 2 1 1 0 1 2 NA NA 2 2 2 NA NA NA NA 1
## [24] 1 0 0 NA 1 NA 0 1 0 2 NA 0 NA 0 0 2 0 NA NA 0 NA NA NA
## [47] NA 1 NA 2
##
## $`49`
## [1] 2 NA 1 NA NA NA 2 0 2 0 0 0 2 0 0 0 NA NA NA 0 NA NA 0
## [24] 2 NA 1 NA 1 NA 0 NA 1 0 1 0 2 2 2 2 2 0 NA 1 NA 2 NA
## [47] NA 0 1 NA
##
## $`50`
## [1] NA 2 1 2 1 NA 2 1 0 NA 0 2 NA 2 NA NA 2 NA NA 1 1 NA NA
## [24] 1 NA 0 2 1 NA 0 NA 1 NA 2 0 NA NA 0 0 0 1 1 2 0 NA 1
## [47] 2 2 0 NA
```

- In one statement, use the `lapply` function to create a list whose keys are the column number and values are themselves a list with keys: “min” whose value is the minimum of the column, “max” whose value is the maximum of the column, “pct_missing” is the proportion of missingness in the column and “first_NA” whose value is the row number of the first time the NA appears. Use the `which` function.

```
list_R_Columns
```

```
## $`1`
## [1] 2 2 NA 1 1 1 NA 2 NA 1 NA 2 NA 1 NA 0 0 2 0 NA NA 1 2
## [24] 0 1 NA 2 0 NA NA NA 2 NA 2 0 0 0 2 2 1 0 0 NA NA 0 1
## [47] NA 0 0 NA
##
## $`2`
## [1] 2 0 1 1 0 0 2 1 NA 0 2 0 0 0 1 NA NA 1 0 0 0 NA NA
## [24] 1 1 0 NA NA NA NA 0 2 0 0 NA NA 2 0 0 NA 2 0 0 0 0 2
## [47] 0 0 1 0
##
## $`3`
## [1] 1 2 2 NA 0 NA NA 2 NA 2 1 0 2 2 2 0 2 0 NA 0 NA 0 2
## [24] NA 1 2 2 1 1 2 2 1 NA 0 2 0 1 1 NA 0 1 0 0 NA 0 1
## [47] 2 1 NA NA
##
## $`4`
## [1] 2 0 NA NA NA 2 2 2 1 NA 0 1 0 1 2 0 1 NA NA 1 1 2 NA
## [24] NA NA NA NA 0 NA NA NA 1 1 2 0 0 0 NA 0 0 2 0 NA 1 0 0
## [47] 0 0 NA NA
##
## $`5`
```

```

## [1] 2 2 1 NA 2 NA 0 1 1 1 0 0 NA 0 0 NA NA NA 1 1 0 1 1
## [24] NA 2 1 0 NA 1 1 NA 0 0 NA NA 0 NA 0 0 NA 0 2 2 NA NA 0
## [47] 0 0 NA 0
##
## $`6`
## [1] 1 NA 2 2 1 2 0 1 2 2 NA 1 1 0 0 1 0 2 1 1 2 0 0
## [24] NA 0 2 2 0 2 0 1 0 NA NA 1 0 NA 2 NA 0 NA 0 0 NA 1 0
## [47] 0 NA 0 NA
##
## $`7`
## [1] 2 1 NA 2 0 0 1 NA 2 2 NA 0 0 0 1 NA 0 0 1 NA 2 0 2
## [24] 2 2 0 0 NA 0 NA 1 0 0 0 NA 0 0 0 1 NA 0 2 1 0 0 NA
## [47] 0 1 NA NA
##
## $`8`
## [1] 1 NA NA 2 NA 0 0 2 NA NA NA NA NA 0 0 0 NA 1 2 0 1 0 1
## [24] NA NA 2 NA 2 0 0 0 0 NA NA NA 0 0 NA 0 0 2 0 1 NA 0 0
## [47] NA 0 0 0
##
## $`9`
## [1] 2 2 1 0 1 0 NA 2 2 NA 0 1 2 1 1 1 2 NA NA NA NA 2 1
## [24] 1 NA 0 NA 0 1 NA 1 0 NA 1 NA NA 1 2 NA NA NA NA 1 NA 0 NA
## [47] NA 0 0 0
##
## $`10`
## [1] 1 NA 1 2 2 2 2 0 NA 0 2 NA NA NA NA NA 0 NA 1 0 2 0 1
## [24] 2 0 1 0 0 0 0 0 0 1 NA NA NA 2 0 2 0 0 NA 0 0 1 0
## [47] NA 1 0 2
##
## $`11`
## [1] 2 1 2 2 2 1 NA 0 0 1 1 1 0 2 0 1 NA 1 1 0 2 0 0
## [24] 0 2 NA 0 0 NA NA NA 0 0 1 NA 0 0 2 2 1 1 1 NA 0 2 NA
## [47] 0 0 1 NA
##
## $`12`
## [1] 0 0 2 NA NA 1 1 NA NA 2 2 2 0 NA 0 1 NA 2 NA 2 0 0 2
## [24] 2 NA NA 0 0 0 NA 2 2 0 1 0 0 NA 0 NA 1 NA 2 2 NA 2 0
## [47] 0 0 NA 2
##
## $`13`
## [1] 0 1 2 NA 0 1 0 0 1 NA 0 2 0 0 NA 2 0 0 0 NA 0 0 2
## [24] NA 2 1 NA NA 0 NA 0 1 2 NA 2 0 NA NA 2 2 0 0 NA 1 1 NA
## [47] NA 2 0 1
##
## $`14`
## [1] NA 1 2 NA 2 0 NA NA 0 NA 2 1 2 0 0 0 0 0 0 NA 1 0 NA
## [24] 1 2 0 2 0 2 0 0 NA NA 0 2 2 0 2 0 NA 0 0 NA NA 0 0
## [47] NA NA 1 0
##
## $`15`
## [1] 1 0 1 1 1 NA 0 0 0 0 NA 1 0 NA 0 NA 2 0 NA NA NA 1 NA
## [24] 0 0 0 2 2 1 NA 1 0 2 0 NA 2 NA 0 2 NA 2 0 1 2 2 0
## [47] 1 NA 2 NA
##

```

```

## $`16`
## [1] 1 1 0 NA 2 0 NA 1 NA 0 2 1 0 2 0 0 0 2 0 NA 0 0 0
## [24] 1 NA NA 0 2 NA NA 0 0 0 0 2 1 0 0 0 1 NA NA 0 1 2 0
## [47] 2 NA 0 0
##
## $`17`
## [1] 0 NA 0 2 2 2 NA NA 1 1 2 0 NA 1 NA 1 NA 2 2 2 0 0 0
## [24] 1 1 0 0 0 2 0 2 0 NA 0 1 0 2 0 NA NA 0 NA NA 0 NA 2
## [47] 0 0 1 NA
##
## $`18`
## [1] NA 1 1 0 2 2 0 2 2 NA NA 2 2 NA 0 2 NA 0 2 1 NA 1 NA
## [24] 0 0 1 0 1 NA NA 0 0 NA 0 NA 0 0 1 NA 1 0 1 NA NA 1 2
## [47] 1 NA NA 0
##
## $`19`
## [1] 0 2 NA NA 0 2 NA 0 0 0 2 0 NA 2 2 NA 0 2 1 0 2 0 NA
## [24] 1 NA 2 NA 0 1 1 0 0 NA NA 1 0 NA 0 0 NA NA 0 0 NA 0 NA
## [47] 1 NA 0 NA
##
## $`20`
## [1] NA 2 0 NA 2 NA NA 0 0 1 NA NA 0 0 0 NA 1 0 NA 2 NA NA 2
## [24] NA 0 1 NA NA NA 0 NA 1 2 0 NA 0 2 NA 2 NA 2 0 0 0 NA 0
## [47] 0 NA NA 0
##
## $`21`
## [1] 0 1 NA NA 0 NA 2 NA 2 2 0 2 NA NA 2 0 NA 0 0 NA NA 2 NA
## [24] NA 0 1 0 NA 1 2 0 2 0 0 NA NA NA 1 NA 2 0 NA 0 0 0 0
## [47] 2 0 NA 0
##
## $`22`
## [1] 2 1 2 2 NA NA 1 2 NA 0 0 0 2 0 0 1 NA NA 2 0 1 1 1
## [24] NA 0 NA 2 NA 0 2 0 0 1 NA 0 1 NA NA 0 NA 1 1 0 0 NA NA
## [47] 2 NA NA 0
##
## $`23`
## [1] NA NA 2 0 1 0 2 NA NA NA 2 0 1 0 1 0 0 1 2 NA NA 0 NA
## [24] 0 NA 0 NA 1 NA 2 NA 0 2 0 0 2 0 0 0 0 0 0 0 1 1 2
## [47] 0 NA NA NA
##
## $`24`
## [1] NA NA 1 NA 1 0 NA 2 0 0 2 NA 0 2 2 0 2 1 2 0 2 0 0
## [24] 2 0 2 0 2 0 0 2 NA NA NA 0 0 0 0 0 NA 1 0 0 2 0 0
## [47] 1 NA NA 0
##
## $`25`
## [1] NA 1 2 0 NA NA 0 2 0 1 0 2 1 0 2 NA 0 NA 0 2 NA 2 NA
## [24] 0 2 NA NA 0 NA 2 1 NA 0 0 0 1 0 1 1 2 1 NA 0 0 NA 0
## [47] 0 2 NA NA
##
## $`26`
## [1] 1 2 NA 1 1 0 0 NA 1 1 0 1 0 NA NA 0 NA 0 NA NA 0 0 NA
## [24] 0 1 0 2 2 0 NA NA 0 2 1 0 0 1 0 0 NA 0 0 0 0 NA 2
## [47] 0 0 0 NA

```

```

##
## $`27`
## [1] 2 NA NA 1 NA NA NA NA NA 0 0 NA 1 2 2 1 1 0 1 0 0 NA 0
## [24] 0 0 2 NA 1 1 NA 0 2 0 2 0 2 0 NA NA 1 2 0 1 0 0 NA
## [47] NA 2 0 NA
##
## $`28`
## [1] 0 0 2 2 0 NA NA 1 0 2 2 NA 1 NA 1 0 2 0 0 0 NA 0 NA
## [24] NA 1 1 NA NA 2 0 2 NA NA NA 0 NA NA NA 0 NA 1 0 2 NA NA NA
## [47] 0 NA NA 0
##
## $`29`
## [1] 2 2 2 NA NA 1 2 2 0 0 1 NA 1 2 1 NA NA 2 0 2 0 NA 1
## [24] 0 NA 1 NA NA 2 2 0 0 0 NA 1 0 0 2 1 1 1 0 0 0 0 NA
## [47] NA NA 0 NA
##
## $`30`
## [1] 2 0 2 2 1 1 NA NA 2 1 NA 0 1 0 1 0 0 0 0 2 1 0 2
## [24] 2 1 0 NA NA NA 0 0 NA 0 2 0 NA 0 0 0 0 1 0 1 NA 0 NA
## [47] 0 0 0 0
##
## $`31`
## [1] NA 2 1 2 NA 0 0 1 1 0 1 2 NA 0 0 0 0 0 1 0 1 NA NA
## [24] 0 0 0 2 0 1 1 NA 1 0 0 1 1 2 0 NA 0 1 NA NA 0 NA NA
## [47] 0 0 1 0
##
## $`32`
## [1] 0 2 0 2 2 NA 0 0 1 1 0 1 1 0 1 2 NA 2 NA NA 1 1 1
## [24] 2 1 0 2 NA 0 2 0 NA 0 2 0 0 0 NA 0 0 1 NA 0 2 0 0
## [47] NA 0 NA 0
##
## $`33`
## [1] 2 0 NA 0 0 1 0 NA 2 NA 1 0 1 NA 2 2 2 1 2 1 NA NA 0
## [24] 1 0 NA 0 0 2 1 NA NA NA NA 0 1 NA 1 NA 1 0 0 1 1 0 0
## [47] 0 0 NA 1
##
## $`34`
## [1] 0 NA 0 NA 2 2 NA 2 NA 0 0 1 1 0 0 1 NA 0 0 NA 2 2 2
## [24] 0 0 1 NA NA 1 2 NA 0 2 1 NA NA 0 2 NA 2 NA NA 0 0 0 NA
## [47] 2 NA 0 1
##
## $`35`
## [1] NA NA 2 1 NA 0 0 0 2 2 1 0 2 0 0 2 2 NA NA NA NA 0 0
## [24] NA 2 NA 0 1 0 2 0 1 0 0 2 0 NA 1 NA 0 NA 1 1 0 1 0
## [47] 0 1 0 NA
##
## $`36`
## [1] 0 1 2 NA NA 1 1 NA 2 1 NA NA 0 2 0 0 0 NA 0 2 1 2 0
## [24] 0 0 0 NA 2 1 0 0 0 2 1 NA NA 1 0 NA 0 NA 0 0 1 NA 1
## [47] 0 0 1 NA
##
## $`37`
## [1] NA 2 NA 0 2 1 0 NA 0 0 1 NA 2 1 NA 2 NA 2 NA NA 0 NA 0
## [24] 0 2 2 NA NA NA 1 2 1 NA NA 0 2 2 0 NA 1 NA 2 1 1 0 0

```



```

## [47] NA NA 1 0
##
## $`38`
## [1] 0 2 0 1 2 0 1 0 0 2 NA 0 NA 0 2 0 2 NA 2 NA 0 2 NA
## [24] NA NA 0 0 NA 0 2 1 0 NA 0 0 2 0 NA NA NA NA 2 NA 1 2 NA
## [47] 0 NA 2 NA
##
## $`39`
## [1] NA 1 NA 0 NA 2 2 0 0 NA 2 2 NA 0 NA 0 1 NA NA 0 NA NA 1
## [24] 1 NA 2 2 NA 0 0 NA 2 1 1 2 NA 1 1 0 0 NA NA 0 2 1 1
## [47] NA 0 1 0
##
## $`40`
## [1] 0 0 NA 0 2 1 2 0 2 1 NA 0 2 0 1 2 0 1 NA NA 0 0 NA
## [24] 2 NA NA 0 0 NA 0 0 1 1 NA 2 1 0 0 0 NA NA 1 1 0 0 NA
## [47] 2 1 2 2
##
## $`41`
## [1] 2 2 0 2 NA 1 2 0 0 1 NA 0 0 1 1 1 2 0 2 2 NA 2 1
## [24] 0 0 0 NA 2 1 0 1 0 NA 2 NA 1 0 0 2 0 0 2 0 NA NA NA
## [47] NA 0 NA NA
##
## $`42`
## [1] 0 0 NA NA 0 0 2 NA 1 0 1 0 NA 2 NA 0 NA NA NA 1 2 2 0
## [24] 0 NA 1 2 0 0 NA 2 1 1 0 2 2 2 NA NA 0 NA NA NA 1 0 0
## [47] NA 1 0 1
##
## $`43`
## [1] 0 1 NA 1 0 1 0 NA NA NA NA 0 NA 1 NA 0 1 2 0 1 0 NA 1
## [24] 1 2 NA 2 1 1 0 NA 1 2 0 2 NA 1 0 1 1 NA 2 1 NA 1 2
## [47] 0 NA NA 0
##
## $`44`
## [1] 2 0 NA 0 1 NA NA NA 1 2 NA 0 NA 2 NA 2 0 NA 0 NA 2 1 NA
## [24] 0 1 NA 0 0 0 0 2 NA 1 1 NA 1 0 0 2 NA NA NA NA 1 0 NA
## [47] NA 1 0 0
##
## $`45`
## [1] 0 NA NA 1 0 2 1 0 0 0 0 NA NA NA NA 2 0 NA 0 NA 1 NA 0
## [24] 0 0 0 NA 0 1 2 NA 2 0 NA 2 NA NA 0 NA 1 NA NA 0 1 0 0
## [47] NA 0 0 1
##
## $`46`
## [1] 2 NA 0 2 NA 2 2 1 0 0 1 NA NA 0 1 NA 2 0 2 2 0 1 NA
## [24] NA NA NA NA 2 1 NA NA NA 0 1 0 NA 2 NA NA 1 NA NA NA 1 0 NA
## [47] 0 0 0 NA
##
## $`47`
## [1] 2 0 0 0 0 2 NA 1 0 0 NA 2 0 1 1 NA NA 0 0 2 NA 1 NA
## [24] 0 NA NA 1 2 1 NA 2 0 1 2 0 NA NA 1 0 1 NA NA 0 NA 0 2
## [47] NA 0 NA NA
##
## $`48`
## [1] NA NA 0 0 NA 0 1 2 1 1 0 1 2 NA NA 2 2 2 NA NA NA NA 1

```

```
## [24] 1 0 0 NA 1 NA 0 1 0 2 NA 0 NA 0 0 2 0 NA NA 0 NA NA NA
## [47] NA 1 NA 2
##
## $`49`
## [1] 2 NA 1 NA NA NA 2 0 2 0 0 0 2 0 0 0 NA NA NA 0 NA NA 0
## [24] 2 NA 1 NA 1 NA 0 NA 1 0 1 0 2 2 2 2 2 0 NA 1 NA 2 NA
## [47] NA 0 1 NA
##
## $`50`
## [1] NA 2 1 2 1 NA 2 1 0 NA 0 2 NA 2 NA NA 2 NA NA 1 1 NA NA
## [24] 1 NA 0 2 1 NA 0 NA 1 NA 2 0 NA NA 0 0 0 1 1 2 0 NA 1
## [47] 2 2 0 NA
```

```
lapply(list_R_Columns, function(x){

  min = min(x, na.rm = TRUE)
  max = max(x, na.rm = TRUE)
  pct_missing = (sum(is.na(x)) / length(x)) * 100
  first_NA = min(which(is.na(x)))

  list("min" = min, "max" = max, "pct_missing" = pct_missing, "first_NA" = first_NA)
} )
```

```
## $`1`
## $`1`$min
## [1] 0
##
## $`1`$max
## [1] 2
##
## $`1`$pct_missing
## [1] 34
##
## $`1`$first_NA
## [1] 3
##
##
## $`2`
## $`2`$min
## [1] 0
##
## $`2`$max
## [1] 2
##
## $`2`$pct_missing
## [1] 24
##
## $`2`$first_NA
## [1] 9
##
##
## $`3`
## $`3`$min
## [1] 0
##
```

```

## $`3`$max
## [1] 2
##
## $`3`$pct_missing
## [1] 24
##
## $`3`$first_NA
## [1] 4
##
##
## $`4`
## $`4`$min
## [1] 0
##
## $`4`$max
## [1] 2
##
## $`4`$pct_missing
## [1] 36
##
## $`4`$first_NA
## [1] 3
##
##
## $`5`
## $`5`$min
## [1] 0
##
## $`5`$max
## [1] 2
##
## $`5`$pct_missing
## [1] 32
##
## $`5`$first_NA
## [1] 4
##
##
## $`6`
## $`6`$min
## [1] 0
##
## $`6`$max
## [1] 2
##
## $`6`$pct_missing
## [1] 22
##
## $`6`$first_NA
## [1] 2
##
##
## $`7`
## $`7`$min

```

```

## [1] 0
##
## $`7`$max
## [1] 2
##
## $`7`$pct_missing
## [1] 24
##
## $`7`$first_NA
## [1] 3
##
##
## $`8`
## $`8`$min
## [1] 0
##
## $`8`$max
## [1] 2
##
## $`8`$pct_missing
## [1] 36
##
## $`8`$first_NA
## [1] 2
##
##
## $`9`
## $`9`$min
## [1] 0
##
## $`9`$max
## [1] 2
##
## $`9`$pct_missing
## [1] 38
##
## $`9`$first_NA
## [1] 7
##
##
## $`10`
## $`10`$min
## [1] 0
##
## $`10`$max
## [1] 2
##
## $`10`$pct_missing
## [1] 26
##
## $`10`$first_NA
## [1] 2
##
##

```

```

## $`11`
## $`11`$min
## [1] 0
##
## $`11`$max
## [1] 2
##
## $`11`$pct_missing
## [1] 20
##
## $`11`$first_NA
## [1] 7
##
##
## $`12`
## $`12`$min
## [1] 0
##
## $`12`$max
## [1] 2
##
## $`12`$pct_missing
## [1] 30
##
## $`12`$first_NA
## [1] 4
##
##
## $`13`
## $`13`$min
## [1] 0
##
## $`13`$max
## [1] 2
##
## $`13`$pct_missing
## [1] 28
##
## $`13`$first_NA
## [1] 4
##
##
## $`14`
## $`14`$min
## [1] 0
##
## $`14`$max
## [1] 2
##
## $`14`$pct_missing
## [1] 28
##
## $`14`$first_NA
## [1] 1

```

```

##
##
## $`15`
## $`15`$min
## [1] 0
##
## $`15`$max
## [1] 2
##
## $`15`$pct_missing
## [1] 28
##
## $`15`$first_NA
## [1] 6
##
##
## $`16`
## $`16`$min
## [1] 0
##
## $`16`$max
## [1] 2
##
## $`16`$pct_missing
## [1] 22
##
## $`16`$first_NA
## [1] 4
##
##
## $`17`
## $`17`$min
## [1] 0
##
## $`17`$max
## [1] 2
##
## $`17`$pct_missing
## [1] 26
##
## $`17`$first_NA
## [1] 2
##
##
## $`18`
## $`18`$min
## [1] 0
##
## $`18`$max
## [1] 2
##
## $`18`$pct_missing
## [1] 32
##
##

```

```

## $`18`$first_NA
## [1] 1
##
##
## $`19`
## $`19`$min
## [1] 0
##
## $`19`$max
## [1] 2
##
## $`19`$pct_missing
## [1] 34
##
## $`19`$first_NA
## [1] 3
##
##
## $`20`
## $`20`$min
## [1] 0
##
## $`20`$max
## [1] 2
##
## $`20`$pct_missing
## [1] 42
##
## $`20`$first_NA
## [1] 1
##
##
## $`21`
## $`21`$min
## [1] 0
##
## $`21`$max
## [1] 2
##
## $`21`$pct_missing
## [1] 36
##
## $`21`$first_NA
## [1] 3
##
##
## $`22`
## $`22`$min
## [1] 0
##
## $`22`$max
## [1] 2
##
## $`22`$pct_missing

```

```

## [1] 32
##
## $`22`$first_NA
## [1] 5
##
##
## $`23`
## $`23`$min
## [1] 0
##
## $`23`$max
## [1] 2
##
## $`23`$pct_missing
## [1] 30
##
## $`23`$first_NA
## [1] 1
##
##
## $`24`
## $`24`$min
## [1] 0
##
## $`24`$max
## [1] 2
##
## $`24`$pct_missing
## [1] 22
##
## $`24`$first_NA
## [1] 1
##
##
## $`25`
## $`25`$min
## [1] 0
##
## $`25`$max
## [1] 2
##
## $`25`$pct_missing
## [1] 30
##
## $`25`$first_NA
## [1] 1
##
##
## $`26`
## $`26`$min
## [1] 0
##
## $`26`$max
## [1] 2

```



```

##
## $`26`$pct_missing
## [1] 26
##
## $`26`$first_NA
## [1] 3
##
##
## $`27`
## $`27`$min
## [1] 0
##
## $`27`$max
## [1] 2
##
## $`27`$pct_missing
## [1] 32
##
## $`27`$first_NA
## [1] 2
##
##
## $`28`
## $`28`$min
## [1] 0
##
## $`28`$max
## [1] 2
##
## $`28`$pct_missing
## [1] 42
##
## $`28`$first_NA
## [1] 6
##
##
## $`29`
## $`29`$min
## [1] 0
##
## $`29`$max
## [1] 2
##
## $`29`$pct_missing
## [1] 28
##
## $`29`$first_NA
## [1] 4
##
##
## $`30`
## $`30`$min
## [1] 0
##
##

```

```

## $`30`$max
## [1] 2
##
## $`30`$pct_missing
## [1] 20
##
## $`30`$first_NA
## [1] 7
##
##
## $`31`
## $`31`$min
## [1] 0
##
## $`31`$max
## [1] 2
##
## $`31`$pct_missing
## [1] 22
##
## $`31`$first_NA
## [1] 1
##
##
## $`32`
## $`32`$min
## [1] 0
##
## $`32`$max
## [1] 2
##
## $`32`$pct_missing
## [1] 20
##
## $`32`$first_NA
## [1] 6
##
##
## $`33`
## $`33`$min
## [1] 0
##
## $`33`$max
## [1] 2
##
## $`33`$pct_missing
## [1] 28
##
## $`33`$first_NA
## [1] 3
##
##
## $`34`
## $`34`$min

```

```

## [1] 0
##
## $`34`$max
## [1] 2
##
## $`34`$pct_missing
## [1] 32
##
## $`34`$first_NA
## [1] 2
##
##
## $`35`
## $`35`$min
## [1] 0
##
## $`35`$max
## [1] 2
##
## $`35`$pct_missing
## [1] 26
##
## $`35`$first_NA
## [1] 1
##
##
## $`36`
## $`36`$min
## [1] 0
##
## $`36`$max
## [1] 2
##
## $`36`$pct_missing
## [1] 26
##
## $`36`$first_NA
## [1] 4
##
##
## $`37`
## $`37`$min
## [1] 0
##
## $`37`$max
## [1] 2
##
## $`37`$pct_missing
## [1] 36
##
## $`37`$first_NA
## [1] 1
##
##

```

```

## `$38`
## `$38`$min
## [1] 0
##
## `$38`$max
## [1] 2
##
## `$38`$pct_missing
## [1] 34
##
## `$38`$first_NA
## [1] 11
##
##
##
## `$39`
## `$39`$min
## [1] 0
##
## `$39`$max
## [1] 2
##
## `$39`$pct_missing
## [1] 34
##
##
## `$39`$first_NA
## [1] 1
##
##
##
## `$40`
## `$40`$min
## [1] 0
##
## `$40`$max
## [1] 2
##
##
## `$40`$pct_missing
## [1] 24
##
##
## `$40`$first_NA
## [1] 3
##
##
##
## `$41`
## `$41`$min
## [1] 0
##
##
## `$41`$max
## [1] 2
##
##
## `$41`$pct_missing
## [1] 24
##
##
## `$41`$first_NA
## [1] 5

```

```

##
##
## $`42`
## $`42`$min
## [1] 0
##
## $`42`$max
## [1] 2
##
## $`42`$pct_missing
## [1] 32
##
## $`42`$first_NA
## [1] 3
##
##
## $`43`
## $`43`$min
## [1] 0
##
## $`43`$max
## [1] 2
##
## $`43`$pct_missing
## [1] 30
##
## $`43`$first_NA
## [1] 3
##
##
## $`44`
## $`44`$min
## [1] 0
##
## $`44`$max
## [1] 2
##
## $`44`$pct_missing
## [1] 38
##
## $`44`$first_NA
## [1] 3
##
##
## $`45`
## $`45`$min
## [1] 0
##
## $`45`$max
## [1] 2
##
## $`45`$pct_missing
## [1] 36
##

```

```

## $`45`$first_NA
## [1] 2
##
##
## $`46`
## $`46`$min
## [1] 0
##
## $`46`$max
## [1] 2
##
## $`46`$pct_missing
## [1] 42
##
## $`46`$first_NA
## [1] 2
##
##
## $`47`
## $`47`$min
## [1] 0
##
## $`47`$max
## [1] 2
##
## $`47`$pct_missing
## [1] 34
##
## $`47`$first_NA
## [1] 7
##
##
## $`48`
## $`48`$min
## [1] 0
##
## $`48`$max
## [1] 2
##
## $`48`$pct_missing
## [1] 40
##
## $`48`$first_NA
## [1] 1
##
##
## $`49`
## $`49`$min
## [1] 0
##
## $`49`$max
## [1] 2
##
## $`49`$pct_missing

```

```
## [1] 36
##
## $`49`$first_NA
## [1] 2
##
##
## $`50`
## $`50`$min
## [1] 0
##
## $`50`$max
## [1] 2
##
## $`50`$pct_missing
## [1] 36
##
## $`50`$first_NA
## [1] 1
```

- Create a vector `v` consisting of a sample of 1,000 iid normal realizations with mean -10 and variance 10.

```
v = rnorm(1000, mean = -10, sd = sqrt(10))
v
```

```
##      [1] -10.1908661 -10.1624923 -11.8462277  -3.2410482 -10.7140202
##      [6] -13.4156861 -15.4484999  -3.1988096 -10.7963862 -12.6025163
##     [11] -13.8471345  -9.4906294  -9.9509346 -14.6981178  -8.9251692
##     [16]  -4.8516204 -10.0171272 -12.4894964  -9.7010786  -8.2105800
##     [21] -11.0378784  -6.5730146  -8.9892636 -10.4921183  -8.3037239
##     [26] -11.6696831 -10.2513837  -8.9509660  -9.7149050  -9.3841297
##     [31]  -8.9613435  -8.2411155 -10.6098050 -10.6084640 -15.7098856
##     [36]  -5.7029345  -9.1984350  -6.6781848  -6.7127765  -9.7193090
##     [41] -12.4129506  -8.1801618  -8.5852421  -8.3539591  -9.4014137
##     [46] -11.5422673 -14.0359000 -15.1369818 -15.6857790 -12.7778382
##     [51] -14.3450922 -11.4856352  -9.0734117  -9.3775780 -11.0338333
##     [56] -12.2249305  -5.7489561  -8.9542430 -12.9092858 -12.4183264
##     [61]  -5.9748369  -9.1329854 -11.3062550 -12.1773119  -6.9962582
##     [66] -10.9259049 -14.4764070  -5.3596342  -9.3126638  -8.5687284
##     [71] -11.2349003  -4.8500193 -12.3412224  -7.2310449  -9.9206584
##     [76]  -8.0504805  -8.8712896  -9.6842984  -7.5195645  -6.5256365
##     [81] -10.2917003 -12.5511990  -7.9427836  -7.8350108 -11.0081415
##     [86]  -6.9575460 -10.2308145  -7.4618764  -5.6834063 -10.1883705
##     [91]  -6.9908555 -13.4221914 -10.6213910 -14.1749027 -14.7855157
##     [96]  -8.2622269 -12.6047268  -8.3174056  -9.9087660  -8.6953497
##    [101]  -8.9472561  -9.2918011  -9.3285724 -15.6671151  -4.5276574
##    [106] -14.7286602  -9.1526224 -11.1093597  -3.9568853  -8.3043443
##    [111] -10.3441418 -11.9638095 -13.8514388 -12.2178674  -5.6367944
##    [116]  -7.8546524 -17.2772558 -13.0327965 -16.8673242  -7.3813365
##    [121]  -8.5763173  -8.4720525 -14.1133859  -8.2602521 -13.7639943
##    [126]  -7.3718924 -14.8255159 -15.4014314  -9.0554812 -11.9799895
##    [131]  -5.7516216 -12.6640186 -10.6956105  -9.6273222  -7.7039627
##    [136]  -8.8031232  -9.2155754  -8.5178796  -8.0292178 -11.1551743
##    [141] -11.7184254  -9.4102988 -10.3199455 -12.1293248  -9.0004887
##    [146] -11.0610560  -9.2483643 -12.4032378  -6.6572070  -8.4042025
##    [151]  -7.7441243 -12.5813479  -5.6106858 -12.9116870 -12.6053956
```

```

## [156] -8.8885504 -6.7970408 -8.4839746 -11.1285608 -6.8544358
## [161] -9.3733907 -11.2132615 -10.9259429 -12.4161430 -1.6809864
## [166] -5.1986973 -10.2780961 -9.6302494 -7.3425107 -15.9561380
## [171] -9.8497334 -6.1311229 -4.4204183 -9.5508416 -12.8627450
## [176] -10.9085406 -12.4800327 -9.5353582 -10.8385721 -8.8655071
## [181] -5.2895839 -12.8983405 -13.2985968 -7.5307267 -2.8868374
## [186] -9.7516790 -5.1098375 -11.4069353 -10.5373264 -11.5894116
## [191] -9.2753008 -12.9784823 -5.9046141 -13.0809267 -12.7574830
## [196] -5.6616236 -9.2838872 -9.6996478 -11.9642942 -18.6836017
## [201] -12.9557940 -5.4522482 -11.6236540 -11.1784022 -12.4472701
## [206] -16.9311754 -16.0554295 -3.6424700 -10.6241707 -8.2293803
## [211] -3.7776529 -7.9737938 -4.3422853 -15.0870100 -8.0781152
## [216] -10.8190227 -6.8262520 -7.8687150 -15.5624168 -14.2403769
## [221] -11.0256468 -7.0665565 -13.7278127 -12.6203150 -8.3355082
## [226] -4.9587303 -4.4408147 -5.4092840 -10.3347714 -4.9064113
## [231] -14.9657553 -6.1379156 -9.2274128 -9.1411422 -10.4377686
## [236] -15.8475988 -9.5971125 -10.9544874 -10.8880315 -5.7538704
## [241] -7.0329473 -10.2225207 -10.8898654 -6.9253904 -7.1645894
## [246] -7.2421692 -7.0785387 -9.7509587 -9.2292059 -8.4432248
## [251] -17.7403598 -8.7587567 -12.4672312 -9.1601349 -5.7030551
## [256] -8.7312050 -13.7370076 -12.3491794 -5.9534547 -14.6990549
## [261] -13.3371021 -7.9601193 -7.9220511 -10.2118609 -13.8667879
## [266] -11.2134227 -11.7749073 -8.9785445 -12.8449022 -15.1462919
## [271] -7.0672215 -9.8113758 -4.0060936 -11.8891934 -8.3565270
## [276] -10.8049283 -15.9403031 -8.3524402 -7.5636454 -10.0754983
## [281] -5.1761777 -5.1736634 -9.2894147 -14.9939929 -8.2529749
## [286] -11.0894640 -13.3309766 -9.6513153 -2.2013678 -11.9730719
## [291] -11.4311755 -9.7453784 -10.3535890 -9.4795773 -12.7733019
## [296] -5.4552507 -9.9013942 -18.4641786 -7.9648376 -9.9727245
## [301] -13.4910020 -7.8097958 -13.3339091 -7.0605575 -10.4160888
## [306] -12.0681951 -8.0407460 -4.9453881 -5.6923005 -6.7787927
## [311] -13.1122883 -16.9079253 -9.8073130 -10.9997902 -8.0298981
## [316] -11.3213608 -8.7699977 -11.9458620 -6.0799959 -12.4206661
## [321] -6.4529860 -5.4783441 -3.7230292 -6.0502665 -8.0496493
## [326] -12.4378421 -7.3249329 -12.5103562 -13.6767457 -7.3773590
## [331] -5.8648797 -12.7692507 -9.5048648 -10.5915455 -12.6195110
## [336] -9.5862443 -13.3874704 -6.3773184 -8.6378618 -11.2567487
## [341] -13.5481535 -7.9514469 -7.7138962 -2.7828980 -10.9841772
## [346] -10.4821419 -11.4613178 -8.8220147 -10.4935506 -6.8694411
## [351] -7.0897455 -7.3146581 -11.3835962 -10.0326047 -4.8365765
## [356] -9.3244574 -15.4066366 -11.3256560 -12.2493466 -9.9926020
## [361] -9.1525336 -8.4941898 -11.7307556 -7.9119686 -12.3432561
## [366] -4.3630736 -12.3304964 -14.2774008 -15.6189318 -6.1873641
## [371] -9.9689718 -8.9145949 -3.0516288 -17.5637684 -9.7467647
## [376] -8.8510685 -15.6944595 -8.5428699 -7.1619674 -13.7737125
## [381] -11.3595621 -9.4628187 -6.9497283 -8.1784831 -17.3207016
## [386] -17.0456211 -5.8123346 -6.7423201 -14.4062425 -7.4944139
## [391] -12.1952715 -13.8049274 -10.6803344 -9.5434339 -12.2336288
## [396] -12.4882623 -14.3397513 1.1334307 -9.8288010 -5.7569618
## [401] -9.7392370 -9.3415990 -12.8012443 -7.2591625 -6.9864168
## [406] -13.2223940 -6.4455472 -8.6508527 -11.9801637 -11.8945595
## [411] -12.1377032 -11.2984734 -7.6902745 0.8763834 -5.3387498
## [416] -9.0434399 -4.2672885 -12.3730596 -11.1622259 -9.7765311
## [421] -2.8702327 -10.6231092 -14.8167857 -5.2375876 -11.6872794

```



```

## [426] -6.5176905 -11.3035248 -10.2842003 -7.6174752 -11.9341522
## [431] -13.4096761 -8.8927783 -13.5015644 -10.0559067 -7.3215854
## [436] -15.0544449 -10.7986755 -4.2664178 -6.7363836 -10.9655015
## [441] -5.5682116 -11.5429867 -9.2270449 -12.4309943 -13.1264329
## [446] -11.4139527 -15.6520255 -11.9081095 -11.9646275 -9.7762971
## [451] -10.2827713 -11.8558689 -13.9142278 -1.7461325 -7.8254117
## [456] -9.7793409 -8.6959891 -10.8684060 -10.7245063 -5.7252580
## [461] -13.7571448 -9.6435968 -6.0900170 -7.3927160 -16.8154412
## [466] -9.4805301 -6.0451956 -8.8772752 -11.9747053 -7.0938255
## [471] -9.5833493 -4.8301902 -5.6395531 -10.6532896 -12.6990119
## [476] -6.3860827 -12.9342066 -11.9917319 -7.6454253 -5.5843525
## [481] -13.6878088 -9.1191027 -11.8655387 -10.9675472 -6.4073252
## [486] -7.4131413 -7.2519960 -4.6398595 -6.4183539 -4.8705039
## [491] -6.7190452 -11.4574161 -9.0855482 -15.4265596 -10.8395146
## [496] -6.2471960 -10.5318569 -15.5668133 -11.3887609 -10.6539067
## [501] -13.2776573 -12.2774337 -8.7056758 -14.7023635 -6.8076132
## [506] -12.2745526 -8.3718777 -12.9941502 -11.2212453 -7.9328274
## [511] -7.0768990 -13.5399414 -6.8843771 -7.2261220 -11.6576369
## [516] -8.0155101 -10.2820279 -9.3286874 -11.1804125 -5.9555637
## [521] -12.4314348 -14.5849660 -12.5087012 -11.5488756 -8.3078760
## [526] -12.5798761 -6.1036780 -8.6811605 -7.4181820 -8.9038415
## [531] -14.6138197 -12.1424940 -16.6398869 -10.8234581 -5.0332345
## [536] -10.8330071 -12.4805634 -12.5803244 -9.4723845 -9.0736586
## [541] -8.8508135 -5.4971515 -12.8348323 -7.8170702 -12.9725734
## [546] -10.6178967 -9.3277409 -15.1063818 -12.2319777 -11.4554010
## [551] -17.0995596 -7.4408630 -7.3617696 -9.5519786 -9.5464829
## [556] -2.4605476 -12.0298618 -10.8663594 -6.0616922 -12.5854355
## [561] -8.6252495 -14.5408473 -7.0332390 -7.0882500 -9.0524208
## [566] -9.6321950 -11.8372288 -15.8718863 -2.0000515 -13.2463098
## [571] -7.5609914 -9.5963835 -9.1825903 -5.6170460 -5.9073014
## [576] -3.3693784 -8.9015588 -14.4978145 -9.3002364 -5.4213829
## [581] -12.9064506 -11.8571231 -11.4896167 -9.2147555 -6.7788110
## [586] -7.1457847 -9.8968656 -10.5253847 -11.2504599 -7.3344045
## [591] -14.0683278 -8.0802558 -7.1615132 -13.5186851 -9.4843048
## [596] -14.7685962 -4.9614740 -13.4536865 -15.4339225 -14.6900398
## [601] -8.0528698 -9.2055008 -9.9436009 -7.5725836 -10.1329388
## [606] -6.8305791 -11.8405745 -10.4891504 -7.8505096 -12.0746190
## [611] -8.3713495 -10.8509805 -18.0420520 -9.8393750 -11.6020391
## [616] -7.4958839 -8.4167026 -13.3438146 -9.7266866 -7.3310105
## [621] -15.1473881 -9.9325850 -12.3804809 -10.9550599 -2.0967686
## [626] -7.6462146 -7.7072425 -8.7445901 -9.3559480 -11.8647290
## [631] -13.7446115 -6.8032004 -13.9582522 -8.8759280 -13.7443192
## [636] -14.1088190 -9.8356486 -11.4197770 -13.9407706 -9.3196978
## [641] -9.1500038 -11.9903915 -8.1614228 -10.4767844 -10.4650199
## [646] -8.3954394 -11.1283831 -9.9179800 -7.5980330 -10.1677139
## [651] -9.7913564 -7.4570626 -13.0010954 -9.9956197 -14.0983660
## [656] -11.8612338 -13.7561718 -7.7945175 -13.1924896 -7.9941650
## [661] -14.7594872 -13.9641120 -6.9052680 -5.6643856 -6.0736446
## [666] -6.5069102 -8.6084666 -4.0799729 -16.1847932 -9.4433171
## [671] -7.8165632 -13.4909462 -4.5865320 -10.4460616 -17.4320575
## [676] -16.1216345 -9.8038753 -11.4701362 -10.5612501 -8.5192398
## [681] -8.5866147 -16.2546668 -9.7408434 -13.1799626 -7.5532667
## [686] -11.2946457 -5.9893724 -3.6165893 -9.8419628 -14.0046864
## [691] -10.0849934 -6.0783224 -13.1266325 -6.4340839 -7.9245088

```

```

## [696] -10.0722602 -3.6427198 -10.5442005 -7.9712162 -11.8352669
## [701] -5.2892050 -11.8518688 -8.3163145 -8.2520560 -10.4716793
## [706] -9.7369203 -12.4722478 -15.0208099 -10.6639277 -12.2412511
## [711] -10.0113729 -7.6491100 -4.9693727 -8.8027588 -10.8604179
## [716] -8.3428640 -9.5924725 -4.8686379 -8.6256077 -7.0527766
## [721] -11.7950439 -9.0809630 -10.8958584 -9.0412652 -10.0848781
## [726] -13.1445539 -8.8197777 -8.8662952 -8.2486043 -11.1086589
## [731] -9.9024039 -17.2578017 -10.1397745 -12.3014259 -12.5602227
## [736] -6.4275944 -9.0997168 -9.0683139 -9.6995862 -7.4274092
## [741] -7.3149472 -1.1152180 -7.5256954 -9.6235494 -15.0717642
## [746] -7.6464076 -8.6044754 -2.1352099 -9.9621398 -6.9403383
## [751] -14.1454347 -15.3007830 -15.1703953 -10.1672631 -11.2835405
## [756] -6.3352942 -9.7121847 -8.5340789 -8.1232506 -14.0483263
## [761] -13.4827661 -9.0334324 -11.6125456 -12.0081797 -9.2313806
## [766] -3.4421451 -9.1382453 -7.8331859 -14.5023813 -5.7612940
## [771] -11.7754224 -9.9484727 -7.8778463 -7.9838595 -9.3737626
## [776] -10.8020722 -15.2720183 -8.0853826 -5.6481532 -7.5449543
## [781] -14.3155885 -9.5670807 -10.7415613 -3.3446396 -3.2852816
## [786] -10.6323968 -10.8093580 -13.7691025 -12.8053353 -10.1691854
## [791] -14.5580737 -9.0764450 -13.1814248 -13.2148389 -9.0733319
## [796] -8.9004754 -9.3786025 -6.5667980 -5.7238144 -11.5065173
## [801] -9.1163624 -9.5969352 -11.3590339 -13.9613516 -10.0521954
## [806] -14.4045244 -8.9177750 -6.0644311 -10.2979177 -10.5500871
## [811] -15.5002218 -6.8425403 -10.5057067 -7.1219862 -15.1895771
## [816] -7.9605382 -10.9585657 -12.5611776 -7.5495507 -7.5335651
## [821] -6.0416292 -10.7293729 -13.8639232 -6.5570047 -8.2476378
## [826] -11.1970109 -4.7616955 -10.8964879 -14.3231682 -13.6319163
## [831] -5.4850235 -12.5253048 -9.0200081 -7.4401108 -10.7448012
## [836] -13.8476181 -9.5865591 -9.3007694 -9.0549806 -14.0805199
## [841] -13.6892781 -6.4202412 -10.8840886 -12.5669734 -13.5840572
## [846] -8.3575565 -10.9148774 -11.9642601 -9.5250492 -6.7742159
## [851] -8.5511282 -7.9739586 -4.7170628 -10.0697310 -10.6074312
## [856] -10.6639298 -10.5168910 -7.3319744 -10.9519592 -7.5637299
## [861] -11.8604542 -9.7266876 -14.2991300 -9.4849424 -14.3724497
## [866] -8.3624448 -14.8506272 -9.9764647 -14.3610446 -12.8769731
## [871] -11.0623259 -9.9020871 -10.8789509 -11.5190788 -12.4771349
## [876] -11.2018280 -10.4459628 -7.6668542 -8.6005680 -5.0551432
## [881] -10.8269585 -8.3989504 -11.2432691 -10.4787924 -11.2236699
## [886] -10.5541020 -7.1301129 -6.7946416 -12.6232073 -5.7130653
## [891] -3.5369311 -9.3504608 -11.5611636 -14.4422526 -11.1017196
## [896] -4.3528089 -13.5989799 -13.0226126 -7.8048083 -14.4666575
## [901] -7.7045640 -8.9513602 -11.0565305 -14.4397486 -8.0099822
## [906] -11.5554995 -12.9978064 -6.6763207 -15.1511435 -15.8413579
## [911] -12.1248239 -16.9108750 -11.3863749 -6.2472386 -13.7108986
## [916] -11.0024647 -10.2875489 -10.4850030 -13.4126176 -8.5844562
## [921] -3.4916605 -11.0165383 -7.6481940 -14.1990295 -7.8195709
## [926] -9.2064388 -14.6242617 -7.4136850 -9.0243262 -12.1911719
## [931] -10.1440250 -14.0509908 -8.7807987 -12.4304351 -15.2653207
## [936] -8.1244307 -11.2780465 -12.8699590 -9.2539594 -4.6467313
## [941] -13.1903132 -13.2081936 -8.5437684 -8.5569407 -7.4833352
## [946] -12.7519743 -10.1624312 -10.7493531 -12.3998804 -12.0850887
## [951] -8.2380500 -10.6228515 -14.5789409 -7.0406329 -4.0067391
## [956] -7.6047008 -10.2368813 -15.5801253 -8.4095242 -13.7342717
## [961] -10.2745182 -11.2254367 -5.6451503 -9.3462194 -17.4887215

```

```
## [966] -11.4098276 -8.0267868 -3.2534837 -13.4440276 -8.6902648
## [971] -9.4341511 -9.1675250 -7.1822473 -14.0408811 -13.0501148
## [976] -7.9724794 -6.7700811 -11.5098280 -10.0543491 -12.4173262
## [981] -7.3537270 -15.3758669 -12.0228078 -9.4438976 -8.4515066
## [986] -10.4958201 -10.5495729 -7.0069196 -6.2269103 -12.6952148
## [991] -16.1497183 -13.3587416 -9.4006023 -7.5353412 -9.8293071
## [996] -10.7057494 -14.3780786 -6.5717664 -10.8430096 -10.7928445
```

- Find the average of `v` and the standard error of `v`.

```
mean(v)
```

```
## [1] -9.969983
```

```
sd(v)/sqrt(length(v))
```

```
## [1] 0.09942725
```

- Find the 5%ile of `v` and use the `qnorm` function to compute what it theoretically should be.

```
quantile(v, 0.05)
```

```
##      5%
```

```
## -15.14758
```

```
qnorm(0.05, mean = -10, sd = sqrt(10), lower.tail = TRUE)
```

```
## [1] -15.20148
```

- Create a list named `my_list` with keys “A”, “B”, ... where the entries are arrays of size 1, 2 x 2, 3 x 3 x 3, etc. Fill the array with the numbers 1, 2, 3, etc. Make 8 entries.

```
keys = c("A", "B", "C", "D", "E", "F", "G", "H")
```

```
my_list=list()
```

```
for(i in 1 : length(keys)){
  my_list[[keys[i]]] = array(seq(1,i), dim = c(rep(i ,i)))
}
```

Test with the following uncomprehensive tests:

```
#expect_equal(my_list$A, 1)
expect_equal(my_list[[2]][, 1], 1 : 2)
expect_equal(dim(my_list[["H"]]), rep(8, 8))
```

Run the following code:

```
lapply(my_list, object.size)
```

```
## $A
## 224 bytes
##
## $B
## 232 bytes
##
## $C
## 352 bytes
##
## $D
## 1248 bytes
##
```

```
## $E
## 12744 bytes
##
## $F
## 186864 bytes
##
## $G
## 3294416 bytes
##
## $H
## 67109104 bytes
```

Use `?object.size` to read about what these functions do. Then explain the output you see above. For the later arrays, does it make sense given the dimensions of the arrays?

Answer here in English. `#object.size` provides an estimation of the amount of memory used to store an object in R. Yes, the increasing dimensions of the arrays = greater amount of memory storage required.

Now cleanup the namespace by deleting all stored objects and functions:

```
?object.size
rm(list = ls())
```

Basic Binary Classification Modeling

- Load the famous `iris` data frame into the namespace. Provide a summary of the columns and write a few descriptive sentences about the distributions using the code below and in English.

```
iris
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width  Species
## 1           5.1         3.5         1.4         0.2    setosa
## 2           4.9         3.0         1.4         0.2    setosa
## 3           4.7         3.2         1.3         0.2    setosa
## 4           4.6         3.1         1.5         0.2    setosa
## 5           5.0         3.6         1.4         0.2    setosa
## 6           5.4         3.9         1.7         0.4    setosa
## 7           4.6         3.4         1.4         0.3    setosa
## 8           5.0         3.4         1.5         0.2    setosa
## 9           4.4         2.9         1.4         0.2    setosa
## 10          4.9         3.1         1.5         0.1    setosa
## 11          5.4         3.7         1.5         0.2    setosa
## 12          4.8         3.4         1.6         0.2    setosa
## 13          4.8         3.0         1.4         0.1    setosa
## 14          4.3         3.0         1.1         0.1    setosa
## 15          5.8         4.0         1.2         0.2    setosa
## 16          5.7         4.4         1.5         0.4    setosa
## 17          5.4         3.9         1.3         0.4    setosa
## 18          5.1         3.5         1.4         0.3    setosa
## 19          5.7         3.8         1.7         0.3    setosa
## 20          5.1         3.8         1.5         0.3    setosa
## 21          5.4         3.4         1.7         0.2    setosa
## 22          5.1         3.7         1.5         0.4    setosa
## 23          4.6         3.6         1.0         0.2    setosa
## 24          5.1         3.3         1.7         0.5    setosa
## 25          4.8         3.4         1.9         0.2    setosa
```

## 26	5.0	3.0	1.6	0.2	setosa
## 27	5.0	3.4	1.6	0.4	setosa
## 28	5.2	3.5	1.5	0.2	setosa
## 29	5.2	3.4	1.4	0.2	setosa
## 30	4.7	3.2	1.6	0.2	setosa
## 31	4.8	3.1	1.6	0.2	setosa
## 32	5.4	3.4	1.5	0.4	setosa
## 33	5.2	4.1	1.5	0.1	setosa
## 34	5.5	4.2	1.4	0.2	setosa
## 35	4.9	3.1	1.5	0.2	setosa
## 36	5.0	3.2	1.2	0.2	setosa
## 37	5.5	3.5	1.3	0.2	setosa
## 38	4.9	3.6	1.4	0.1	setosa
## 39	4.4	3.0	1.3	0.2	setosa
## 40	5.1	3.4	1.5	0.2	setosa
## 41	5.0	3.5	1.3	0.3	setosa
## 42	4.5	2.3	1.3	0.3	setosa
## 43	4.4	3.2	1.3	0.2	setosa
## 44	5.0	3.5	1.6	0.6	setosa
## 45	5.1	3.8	1.9	0.4	setosa
## 46	4.8	3.0	1.4	0.3	setosa
## 47	5.1	3.8	1.6	0.2	setosa
## 48	4.6	3.2	1.4	0.2	setosa
## 49	5.3	3.7	1.5	0.2	setosa
## 50	5.0	3.3	1.4	0.2	setosa
## 51	7.0	3.2	4.7	1.4	versicolor
## 52	6.4	3.2	4.5	1.5	versicolor
## 53	6.9	3.1	4.9	1.5	versicolor
## 54	5.5	2.3	4.0	1.3	versicolor
## 55	6.5	2.8	4.6	1.5	versicolor
## 56	5.7	2.8	4.5	1.3	versicolor
## 57	6.3	3.3	4.7	1.6	versicolor
## 58	4.9	2.4	3.3	1.0	versicolor
## 59	6.6	2.9	4.6	1.3	versicolor
## 60	5.2	2.7	3.9	1.4	versicolor
## 61	5.0	2.0	3.5	1.0	versicolor
## 62	5.9	3.0	4.2	1.5	versicolor
## 63	6.0	2.2	4.0	1.0	versicolor
## 64	6.1	2.9	4.7	1.4	versicolor
## 65	5.6	2.9	3.6	1.3	versicolor
## 66	6.7	3.1	4.4	1.4	versicolor
## 67	5.6	3.0	4.5	1.5	versicolor
## 68	5.8	2.7	4.1	1.0	versicolor
## 69	6.2	2.2	4.5	1.5	versicolor
## 70	5.6	2.5	3.9	1.1	versicolor
## 71	5.9	3.2	4.8	1.8	versicolor
## 72	6.1	2.8	4.0	1.3	versicolor
## 73	6.3	2.5	4.9	1.5	versicolor
## 74	6.1	2.8	4.7	1.2	versicolor
## 75	6.4	2.9	4.3	1.3	versicolor
## 76	6.6	3.0	4.4	1.4	versicolor
## 77	6.8	2.8	4.8	1.4	versicolor
## 78	6.7	3.0	5.0	1.7	versicolor
## 79	6.0	2.9	4.5	1.5	versicolor

## 80	5.7	2.6	3.5	1.0 versicolor
## 81	5.5	2.4	3.8	1.1 versicolor
## 82	5.5	2.4	3.7	1.0 versicolor
## 83	5.8	2.7	3.9	1.2 versicolor
## 84	6.0	2.7	5.1	1.6 versicolor
## 85	5.4	3.0	4.5	1.5 versicolor
## 86	6.0	3.4	4.5	1.6 versicolor
## 87	6.7	3.1	4.7	1.5 versicolor
## 88	6.3	2.3	4.4	1.3 versicolor
## 89	5.6	3.0	4.1	1.3 versicolor
## 90	5.5	2.5	4.0	1.3 versicolor
## 91	5.5	2.6	4.4	1.2 versicolor
## 92	6.1	3.0	4.6	1.4 versicolor
## 93	5.8	2.6	4.0	1.2 versicolor
## 94	5.0	2.3	3.3	1.0 versicolor
## 95	5.6	2.7	4.2	1.3 versicolor
## 96	5.7	3.0	4.2	1.2 versicolor
## 97	5.7	2.9	4.2	1.3 versicolor
## 98	6.2	2.9	4.3	1.3 versicolor
## 99	5.1	2.5	3.0	1.1 versicolor
## 100	5.7	2.8	4.1	1.3 versicolor
## 101	6.3	3.3	6.0	2.5 virginica
## 102	5.8	2.7	5.1	1.9 virginica
## 103	7.1	3.0	5.9	2.1 virginica
## 104	6.3	2.9	5.6	1.8 virginica
## 105	6.5	3.0	5.8	2.2 virginica
## 106	7.6	3.0	6.6	2.1 virginica
## 107	4.9	2.5	4.5	1.7 virginica
## 108	7.3	2.9	6.3	1.8 virginica
## 109	6.7	2.5	5.8	1.8 virginica
## 110	7.2	3.6	6.1	2.5 virginica
## 111	6.5	3.2	5.1	2.0 virginica
## 112	6.4	2.7	5.3	1.9 virginica
## 113	6.8	3.0	5.5	2.1 virginica
## 114	5.7	2.5	5.0	2.0 virginica
## 115	5.8	2.8	5.1	2.4 virginica
## 116	6.4	3.2	5.3	2.3 virginica
## 117	6.5	3.0	5.5	1.8 virginica
## 118	7.7	3.8	6.7	2.2 virginica
## 119	7.7	2.6	6.9	2.3 virginica
## 120	6.0	2.2	5.0	1.5 virginica
## 121	6.9	3.2	5.7	2.3 virginica
## 122	5.6	2.8	4.9	2.0 virginica
## 123	7.7	2.8	6.7	2.0 virginica
## 124	6.3	2.7	4.9	1.8 virginica
## 125	6.7	3.3	5.7	2.1 virginica
## 126	7.2	3.2	6.0	1.8 virginica
## 127	6.2	2.8	4.8	1.8 virginica
## 128	6.1	3.0	4.9	1.8 virginica
## 129	6.4	2.8	5.6	2.1 virginica
## 130	7.2	3.0	5.8	1.6 virginica
## 131	7.4	2.8	6.1	1.9 virginica
## 132	7.9	3.8	6.4	2.0 virginica
## 133	6.4	2.8	5.6	2.2 virginica

```
## 134      6.3      2.8      5.1      1.5 virginica
## 135      6.1      2.6      5.6      1.4 virginica
## 136      7.7      3.0      6.1      2.3 virginica
## 137      6.3      3.4      5.6      2.4 virginica
## 138      6.4      3.1      5.5      1.8 virginica
## 139      6.0      3.0      4.8      1.8 virginica
## 140      6.9      3.1      5.4      2.1 virginica
## 141      6.7      3.1      5.6      2.4 virginica
## 142      6.9      3.1      5.1      2.3 virginica
## 143      5.8      2.7      5.1      1.9 virginica
## 144      6.8      3.2      5.9      2.3 virginica
## 145      6.7      3.3      5.7      2.5 virginica
## 146      6.7      3.0      5.2      2.3 virginica
## 147      6.3      2.5      5.0      1.9 virginica
## 148      6.5      3.0      5.2      2.0 virginica
## 149      6.2      3.4      5.4      2.3 virginica
## 150      5.9      3.0      5.1      1.8 virginica
```

#There are five columns, four of which are numerical variables (sepal length & width, petal length & width)

The outcome metric is **Species**. This is what we will be trying to predict. However, we have only done binary classification in class (i.e. two classes). Thus the first order of business is to drop one class. Let's drop the level "virginica" from the data frame.

```
virginica_Rows = which(iris$Species == "virginica")
iris2 = iris[-(virginica_Rows), ]
iris2
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width  Species
## 1      5.1      3.5      1.4      0.2      setosa
## 2      4.9      3.0      1.4      0.2      setosa
## 3      4.7      3.2      1.3      0.2      setosa
## 4      4.6      3.1      1.5      0.2      setosa
## 5      5.0      3.6      1.4      0.2      setosa
## 6      5.4      3.9      1.7      0.4      setosa
## 7      4.6      3.4      1.4      0.3      setosa
## 8      5.0      3.4      1.5      0.2      setosa
## 9      4.4      2.9      1.4      0.2      setosa
## 10     4.9      3.1      1.5      0.1      setosa
## 11     5.4      3.7      1.5      0.2      setosa
## 12     4.8      3.4      1.6      0.2      setosa
## 13     4.8      3.0      1.4      0.1      setosa
## 14     4.3      3.0      1.1      0.1      setosa
## 15     5.8      4.0      1.2      0.2      setosa
## 16     5.7      4.4      1.5      0.4      setosa
## 17     5.4      3.9      1.3      0.4      setosa
## 18     5.1      3.5      1.4      0.3      setosa
## 19     5.7      3.8      1.7      0.3      setosa
## 20     5.1      3.8      1.5      0.3      setosa
## 21     5.4      3.4      1.7      0.2      setosa
## 22     5.1      3.7      1.5      0.4      setosa
## 23     4.6      3.6      1.0      0.2      setosa
## 24     5.1      3.3      1.7      0.5      setosa
## 25     4.8      3.4      1.9      0.2      setosa
## 26     5.0      3.0      1.6      0.2      setosa
## 27     5.0      3.4      1.6      0.4      setosa
```

## 28	5.2	3.5	1.5	0.2	setosa
## 29	5.2	3.4	1.4	0.2	setosa
## 30	4.7	3.2	1.6	0.2	setosa
## 31	4.8	3.1	1.6	0.2	setosa
## 32	5.4	3.4	1.5	0.4	setosa
## 33	5.2	4.1	1.5	0.1	setosa
## 34	5.5	4.2	1.4	0.2	setosa
## 35	4.9	3.1	1.5	0.2	setosa
## 36	5.0	3.2	1.2	0.2	setosa
## 37	5.5	3.5	1.3	0.2	setosa
## 38	4.9	3.6	1.4	0.1	setosa
## 39	4.4	3.0	1.3	0.2	setosa
## 40	5.1	3.4	1.5	0.2	setosa
## 41	5.0	3.5	1.3	0.3	setosa
## 42	4.5	2.3	1.3	0.3	setosa
## 43	4.4	3.2	1.3	0.2	setosa
## 44	5.0	3.5	1.6	0.6	setosa
## 45	5.1	3.8	1.9	0.4	setosa
## 46	4.8	3.0	1.4	0.3	setosa
## 47	5.1	3.8	1.6	0.2	setosa
## 48	4.6	3.2	1.4	0.2	setosa
## 49	5.3	3.7	1.5	0.2	setosa
## 50	5.0	3.3	1.4	0.2	setosa
## 51	7.0	3.2	4.7	1.4	versicolor
## 52	6.4	3.2	4.5	1.5	versicolor
## 53	6.9	3.1	4.9	1.5	versicolor
## 54	5.5	2.3	4.0	1.3	versicolor
## 55	6.5	2.8	4.6	1.5	versicolor
## 56	5.7	2.8	4.5	1.3	versicolor
## 57	6.3	3.3	4.7	1.6	versicolor
## 58	4.9	2.4	3.3	1.0	versicolor
## 59	6.6	2.9	4.6	1.3	versicolor
## 60	5.2	2.7	3.9	1.4	versicolor
## 61	5.0	2.0	3.5	1.0	versicolor
## 62	5.9	3.0	4.2	1.5	versicolor
## 63	6.0	2.2	4.0	1.0	versicolor
## 64	6.1	2.9	4.7	1.4	versicolor
## 65	5.6	2.9	3.6	1.3	versicolor
## 66	6.7	3.1	4.4	1.4	versicolor
## 67	5.6	3.0	4.5	1.5	versicolor
## 68	5.8	2.7	4.1	1.0	versicolor
## 69	6.2	2.2	4.5	1.5	versicolor
## 70	5.6	2.5	3.9	1.1	versicolor
## 71	5.9	3.2	4.8	1.8	versicolor
## 72	6.1	2.8	4.0	1.3	versicolor
## 73	6.3	2.5	4.9	1.5	versicolor
## 74	6.1	2.8	4.7	1.2	versicolor
## 75	6.4	2.9	4.3	1.3	versicolor
## 76	6.6	3.0	4.4	1.4	versicolor
## 77	6.8	2.8	4.8	1.4	versicolor
## 78	6.7	3.0	5.0	1.7	versicolor
## 79	6.0	2.9	4.5	1.5	versicolor
## 80	5.7	2.6	3.5	1.0	versicolor
## 81	5.5	2.4	3.8	1.1	versicolor


```
## 82      5.5      2.4      3.7      1.0 versicolor
## 83      5.8      2.7      3.9      1.2 versicolor
## 84      6.0      2.7      5.1      1.6 versicolor
## 85      5.4      3.0      4.5      1.5 versicolor
## 86      6.0      3.4      4.5      1.6 versicolor
## 87      6.7      3.1      4.7      1.5 versicolor
## 88      6.3      2.3      4.4      1.3 versicolor
## 89      5.6      3.0      4.1      1.3 versicolor
## 90      5.5      2.5      4.0      1.3 versicolor
## 91      5.5      2.6      4.4      1.2 versicolor
## 92      6.1      3.0      4.6      1.4 versicolor
## 93      5.8      2.6      4.0      1.2 versicolor
## 94      5.0      2.3      3.3      1.0 versicolor
## 95      5.6      2.7      4.2      1.3 versicolor
## 96      5.7      3.0      4.2      1.2 versicolor
## 97      5.7      2.9      4.2      1.3 versicolor
## 98      6.2      2.9      4.3      1.3 versicolor
## 99      5.1      2.5      3.0      1.1 versicolor
## 100     5.7      2.8      4.1      1.3 versicolor
```

Now create a vector `y` that is length the number of remaining rows in the data frame whose entries are 0 if “setosa” and 1 if “versicolor”.

```
y = ifelse(iris2$Species == "versicolor", 1, 0)
y
```

```
## [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [36] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [71] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

- Fit a threshold model to `y` using the feature `Sepal.Length`. Try to write your own code to do this. What is the estimated value of the threshold parameter? What is the total number of errors this model makes?

```
sepal_Lengths = iris2$Sepal.Length
n = length(sepal_Lengths)
p = ncol(iris2)

w = rep(NA, nrow(iris2))

if(is.null(w)){
  w = rep(0, (p+1))
}

# sepal_Length_2_Species = cbind(x,y)
# sepal_Length_2_Species
#
# for(i in 1:p){
#   for (j in 1:n){
#     x_i = Xinput[i, ]
#     yhat_i = ifelse(sum(x_i *w) > 0, 1, 0)
#     y_i = y_binary[i]
#
#     for(k in 1:(p+1)){
#
#       w[j] = w[j] + (y_i - yhat_i) * x_i[j]
```

```
# }  
# }  
# }
```

Does this make sense given the following summaries:

```
summary(iris[iris$Species == "setosa", "Sepal.Length"])
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
##    4.300   4.800   5.000   5.006   5.200   5.800
```

```
summary(iris[iris$Species == "virginica", "Sepal.Length"])
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
##    4.900   6.225   6.500   6.588   6.900   7.900
```

Write your answer here in English.

TO-DO

- What is the total number of errors this model makes (in-sample)?

```
#TO-DO
```