HW01p

Pizon Shetu

February 24, 2018

Welcome to HW01p where the "p" stands for "practice" meaning you will use R to solve practical problems. This homework is due 11:59 PM Satuday 2/24/18.

You should have RStudio installed to edit this file. You will write code in places marked "TO-DO" to complete the problems. Some of this will be a pure programming assignment. The tools for the solutions to these problems can be found in the class practice lectures. I want you to use the methods I taught you, not for you to google and come up with whatever works. You won't learn that way.

To "hand in" the homework, you should compile or publish this file into a PDF that includes output of your code. Once it's done, push by the deadline.

R Basics

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.

```
pacman:: p_load("testthat")
if (!require("pacman")){install.packages("pacman")}
```

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

```
pacman::p_load(devtools)
pacman::p_load_gh("hadley/ggplot2")
```

1. 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
```

```
#TO-DO
```

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.

2. 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).

```
v = seq(1, 10)
my_reverse =   v[length(v):1]
my_reverse
```

```
##  [1] 10  9  8  7  6  5  4  3  2  1
```

```
my_reverse =  function(v){
  v[length(v):1]
}
my_reverse
```

```
## function(v){
##   v[length(v):1]
## }
```

```
#head(v)
#tail(v)
#TO-DO
```

Test using the following code:

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

3. Let `n = 50`. Create a nxn matrix `R` of exactly 50% entries 0's, 25% 1's 25% 2's in random locations.

```
n = 50
R = matrix(sample(c(rep(0, 1250),rep(1, 625),rep(2, 625)),
size = n*n,
replace = FALSE),
nrow = n,
ncol = n)
R
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,]    0    2    0    2    2    1    0    0    0     0     0     0     0
##  [2,]    2    0    0    2    0    1    2    1    2     0     1     2     2
##  [3,]    1    0    2    2    1    1    2    2    2     0     2     2     1
##  [4,]    1    0    0    2    0    2    1    0    0     0     0     2     0
##  [5,]    1    2    1    0    2    1    0    1    2     2     0     1     2
##  [6,]    1    2    0    0    0    0    0    1    0     0     2     2     2
##  [7,]    2    0    1    0    0    0    1    1    0     0     0     1     1
##  [8,]    1    0    0    2    0    0    0    0    0     0     2     0     1
##  [9,]    2    0    0    2    0    0    2    1    1     1     0     0     1
## [10,]    2    0    0    0    1    2    0    0    1     0     1     2     2
## [11,]    0    0    0    0    0    0    0    0    1     1     2     1     1
## [12,]    0    1    0    2    2    1    0    0    0     2     0     1     0
## [13,]    2    0    1    0    1    1    0    2    1     0     1     0     2
## [14,]    0    0    0    0    2    1    0    0    0     1     1     0     1
```

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

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

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

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

```
## [27,]    2    2    1    1
## [28,]    0    1    1    1
## [29,]    2    0    1    0
## [30,]    1    2    1    2
## [31,]    2    0    1    2
## [32,]    0    0    0    2
## [33,]    2    0    0    1
## [34,]    0    1    2    0
## [35,]    1    0    0    0
## [36,]    0    0    1    0
## [37,]    1    0    0    0
## [38,]    1    2    2    1
## [39,]    0    0    1    0
## [40,]    0    2    0    2
## [41,]    0    2    0    0
## [42,]    0    0    0    1
## [43,]    0    0    1    0
## [44,]    0    0    0    1
## [45,]    1    0    0    0
## [46,]    0    1    0    2
## [47,]    0    2    0    2
## [48,]    0    0    0    0
## [49,]    0    0    2    2
## [50,]    2    0    0    0
```

```
#x = sample(rep(c(0, 1), 50))


#TO-DO
```

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

```
expect_equal(dim(R), c(n, n))
#TO-DO test that the only unique values are 0, 1, 2
#TO-DO test that there are exactly 625 2's
```

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

```
# SO i didn't know how to input a function where it replaced values with NA's so I did the math where 3
n = 50
R = matrix(sample(c(rep(0, 875),rep(1, 438),rep(2, 437), rep(NA, 750)),
size = n*n,
replace = FALSE),
nrow = n,
ncol = n)
R
```

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

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

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

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

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

```
## [20,]    NA    NA     0     0
## [21,]     0     2    NA    NA
## [22,]    NA     1     0     2
## [23,]    NA    NA     0    NA
## [24,]     2    NA     1     1
## [25,]     0     0    NA    NA
## [26,]     2     1    NA     0
## [27,]    NA     2     0     0
## [28,]     0     1     0     0
## [29,]     0     0     0     0
## [30,]     0     1     1     0
## [31,]     1     0     0     2
## [32,]     0     0    NA     0
## [33,]     2     0     2    NA
## [34,]     0     1     0     2
## [35,]     0    NA     0     0
## [36,]     0     0     1    NA
## [37,]     2     2     0     2
## [38,]    NA     1    NA     0
## [39,]    NA     0    NA     2
## [40,]     0     0     0    NA
## [41,]    NA     1    NA    NA
## [42,]    NA     2     0    NA
## [43,]     0     0     0     2
## [44,]     0     0     0     0
## [45,]     0    NA     0     1
## [46,]     2     2     0    NA
## [47,]    NA     0     2     1
## [48,]     2     2    NA     0
## [49,]    NA     1     0     1
## [50,]    NA     0    NA    NA
```

```
#TO-DO
```

Test using the following code. Note this test may fail 1/100 times.

```
num_missing_in_R = sum(is.na(c(R)))
expect_lt(num_missing_in_R, qbinom(0.995, n^2, 0.3))
expect_gt(num_missing_in_R, qbinom(0.005, n^2, 0.3))
```

5. Sort the rows matrix `R` by the largest row sum to lowest. See 2/3 way through practice lecture 3 for a hint.

```
R <- R[order(-rowSums(R, na.rm=TRUE)), ]

# for (i in 1 : nrow(R)){
#   sort(sum(R[i,], na.rm = F), decreasing = T)
# print(R)
# }
# Had a hard time with this problem trying to figure out how to check the sum of each row and then sort
#TO-DO
```

Test using the following code.

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

```
}
```

6. 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] -14.6294241  -9.9145836  -9.6619379 -14.0620263  -7.7227669
##    [6] -11.9562474 -13.5882949  -9.9824140  -9.7228993  -6.7800324
##   [11] -17.5040035 -10.8767312 -12.1199032  -9.6964134  -6.2452054
##   [16] -10.8349699  -5.4377423  -7.9169185 -12.8816012  -8.6297727
##   [21] -15.2376587  -5.6917722  -9.1749137 -13.9784297 -10.2822475
##   [26]  -9.8539316 -11.8150023  -7.2056322 -12.9201035  -9.1091724
##   [31]  -9.2550050  -4.5815529 -18.9085581 -11.0438062  -5.2316432
##   [36] -11.6411550  -6.6284901  -7.3178787  -2.2986721 -10.9955805
##   [41] -10.3553865 -15.7218886  -9.7192509  -9.8983709  -5.4659921
##   [46]  -6.7577596  -9.5362776  -7.3494983 -12.4020673 -13.2982195
##   [51] -13.0448031 -10.4512662 -14.6274783 -13.1187817 -11.6924576
##   [56]  -8.3038718 -11.2345211  -7.5937812 -11.9282692  -9.4336764
##   [61]  -7.4702252 -10.2325732 -18.5670989 -10.2757784 -11.5609788
##   [66]  -6.6457696 -12.7786377 -13.8517731 -11.8879323  -9.4448434
##   [71]  -9.4656820  -8.9488231  -9.6069001 -17.0128953 -15.9357881
##   [76]  -9.0610686  -8.2687854  -9.0748665  -5.5181284  -6.9551631
##   [81] -10.4934875  -9.5849472  -9.4939643 -16.5009709  -5.8300413
##   [86] -10.9853989 -13.9361933  -2.6262906 -10.6430982 -11.9415140
##   [91] -11.3189713  -9.5280506  -9.6059108  -4.7396566  -5.2711352
##   [96]  -6.5313823  -6.6166208  -8.8868842  -8.8979762 -12.1234891
##  [101]  -8.7650539  -9.8768493 -11.3965902 -10.0241970  -0.1720633
##  [106] -11.9411555 -11.3386393 -17.6510203 -11.8029454 -10.3519396
##  [111]  -7.9490876  -9.7849560  -4.5030064  -4.8195673 -13.5154323
##  [116] -10.0916167  -8.6717815  -9.6027514 -15.2367327 -10.9749315
##  [121] -12.5921346  -6.7447022 -10.1876378 -10.7149864  -6.1589038
##  [126]  -5.7289158  -7.9076628 -12.8433902 -12.5029269  -3.3646265
##  [131]  -5.8607410 -10.8648504  -8.6367062 -12.9215542  -5.1342923
##  [136]  -6.7203799  -4.9384842 -14.9998663  -5.4168928 -15.2931118
##  [141] -11.3206311  -9.7596439 -11.2500391 -10.8667464 -10.5943190
##  [146]  -8.7896936  -7.8416429 -10.8285401  -6.7460872  -4.1922218
##  [151] -12.0419478 -13.2361841  -9.3463902 -14.3909435  -3.4646694
##  [156] -12.4597394  -8.1850504  -7.6738284  -7.6614191 -13.2765363
##  [161] -10.4383213  -9.3226681  -9.0031269  -8.2422241 -11.0520517
##  [166]  -8.9975696 -11.4980117  -8.5026211 -13.2766761 -12.4590652
##  [171]  -9.0786808  -7.2973742 -12.4737956  -6.6011357 -12.0061406
##  [176]  -8.5591196 -13.5334893 -11.4243729 -10.5968698  -7.1556402
##  [181]  -7.3894634 -10.3287341  -7.0618062  -8.5777308  -7.5232980
##  [186]  -7.8067106  -7.8150862 -10.0463449  -4.9562468  -8.0986063
##  [191]  -9.7045447  -8.3185072  -3.3394967 -10.1174945  -7.7763242
##  [196]  -9.3485201  -9.2131643  -6.2584789 -12.7461100 -18.2430179
##  [201] -14.0148910 -15.6721820 -12.4598617  -6.0648315 -11.5837997
##  [206]  -6.9921822 -10.9323059 -14.0651750 -10.8030761  -2.1887788
##  [211] -14.1329122 -11.2403145  -7.3982389 -10.0256551  -8.7830270
##  [216]  -9.8872843 -10.4855692  -6.7101295  -2.3156266 -12.0402025
##  [221] -13.7781500  -9.0496883  -8.0361889 -11.3586304  -9.9843568
##  [226]  -4.4650259 -12.9473816  -8.6417729  -7.8063482  -3.9879281
##  [231] -13.2148305  -6.2049678  -5.9367739 -10.0907139 -10.1901048
##  [236]  -8.7262393 -10.7257481  -6.0832834  -7.6846637  -9.2698902
```

```
##  [241]  -5.2942539 -13.5431921  -3.0875876 -10.1630436  -6.8993441
##  [246] -11.5747536 -13.5619455 -10.7403740 -13.6252597 -10.6594288
##  [251]  -9.1124429  -8.3010354 -12.8670013 -10.3360758 -21.2613846
##  [256] -15.8062884  -9.8016492  -9.8450612  -9.1156377 -13.0263666
##  [261] -12.6518586 -13.4571693  -3.0790365  -7.3864032  -8.9740134
##  [266] -15.5018287 -11.0811116  -5.8918934 -16.7269332 -11.2454064
##  [271] -12.0833114 -12.3216889 -11.9489553  -8.2343236 -11.7792272
##  [276] -10.2761487 -11.6241472  -7.6406648  -9.3397340 -13.8981215
##  [281] -14.0033253  -6.6312546  -6.1975529 -12.6111686 -12.9767680
##  [286]  -9.3411810 -12.7611879 -12.6204714  -8.7672898 -11.9498833
##  [291] -13.0894898  -6.5438041 -10.8778556  -6.3111894 -11.9350232
##  [296] -14.5039855  -5.5792247 -13.7950177 -11.3566689  -8.7125969
##  [301] -10.2592855 -10.3168023 -15.7103226  -6.1523522  -4.6714583
##  [306]  -8.6037284  -9.8844439 -12.5078166  -8.8120233  -9.6234079
##  [311]  -6.2108823  -9.9539304 -13.8819496  -8.8432671 -13.6331695
##  [316] -16.8134272 -11.6094567 -11.1185627 -12.7812010  -6.3814596
##  [321] -15.9349821  -7.1452102  -9.3394240  -8.2913398  -7.2843373
##  [326]  -9.1957682  -9.0782880  -8.7760200 -15.6662756  -9.9792362
##  [331] -10.2571107 -10.0533887  -6.6209021  -6.3104090  -6.1431318
##  [336]  -3.4738235 -13.5304749  -5.8796040 -11.7803990 -16.1542213
##  [341]  -8.8506700 -12.5739836  -2.8288075 -12.1268149  -3.3045714
##  [346] -11.2782994  -9.5743971 -12.6437385  -7.8171440  -9.7147165
##  [351] -12.8735065 -14.7188786  -9.1179850  -8.7672444  -5.7081444
##  [356] -10.3023685 -12.2123886  -8.2783234 -17.4096502  -9.6405336
##  [361] -12.4745848  -9.1940963  -9.0228678 -12.2464517  -4.6251398
##  [366] -10.6253854  -9.2807847  -8.9275607  -1.8880783 -12.3836052
##  [371]  -7.6821925 -11.6825532  -2.0896513  -6.5332228 -11.8969800
##  [376] -10.4483274  -5.6801967 -11.2161340 -12.6931468 -12.4052252
##  [381]  -6.7542706  -9.9432349  -7.2724475 -14.5271219 -15.7217452
##  [386]  -9.9440344 -10.0406605 -14.6213777 -11.2460688 -10.0273065
##  [391] -12.6265695 -13.3443699  -9.2600416  -8.4242250 -12.6279222
##  [396] -13.0507268  -6.8178145  -9.6432761 -12.6106916 -13.1322883
##  [401] -11.3241377 -11.6228137  -6.8754262  -6.4382872  -7.7099744
##  [406]  -6.6475449 -14.6451479 -11.8935346 -12.7951823 -12.7707225
##  [411] -13.3980499 -11.7506262 -12.3041963 -13.8180081  -8.0582038
##  [416]  -1.4304721 -11.5678760 -10.5256169 -10.5951131  -9.6288197
##  [421] -11.7940301 -15.0146248 -10.8881716  -8.2175784  -8.5646292
##  [426]  -8.5836802  -7.7653448  -9.5784990 -12.2603595 -12.4464607
##  [431] -11.3864303 -10.2538652 -15.9527778 -10.4444734  -9.1240200
##  [436]  -8.8619167  -8.0770478 -11.9526358  -9.6202516 -12.6779886
##  [441] -12.0711183  -3.5689414 -11.1867417  -4.3065347 -10.9511753
##  [446] -10.0093603  -8.3887831  -6.0156563 -13.2305372  -6.3106679
##  [451] -13.5172351 -10.1992660  -4.3961859  -7.0264399 -17.1644842
##  [456] -10.5531411  -8.4001882 -10.8662964 -10.2361427 -12.5324820
##  [461] -10.6870545  -8.4629641 -12.8149588 -10.7592543  -5.7110152
##  [466]  -6.9215311  -6.2752842 -20.2084701 -12.7883786  -9.2857033
##  [471] -11.2961811  -7.4740892 -10.6764554  -6.8332045 -15.1645610
##  [476] -11.7774850  -7.8308241  -9.1048821  -9.7041765 -11.2563846
##  [481]  -7.9743744  -7.2183044  -8.5054808  -6.7750206  -7.6993101
##  [486] -14.4184078 -12.0913665  -9.0570960 -10.2709700 -10.6810955
##  [491] -10.1689780 -11.6436397 -13.1104126 -10.1005877  -6.2502157
##  [496]  -0.6152095  -9.6368640 -10.8441057  -7.4494416 -14.5958754
##  [501] -11.5428448  -6.1722870  -8.7514694  -8.7782788 -13.1375467
##  [506] -12.6086985 -10.0464832  -6.5946772 -15.1541562 -12.8039546
```

```
##  [511] -10.0807946 -19.1735191  -8.6574947 -11.2021311  -7.8429999
##  [516]  -9.5484555 -12.0131630 -15.6495256  -2.6123011  -9.6385687
##  [521]  -7.5954281 -10.0473373 -12.0916583 -11.0529273 -11.6614210
##  [526] -11.2336141 -11.5904606 -11.1313350  -8.3651007 -12.2228173
##  [531]  -9.7824705  -9.0646443 -13.1990582 -10.1038753 -15.9832977
##  [536] -11.2374792 -13.7007422 -11.0031832 -10.7660678 -10.8860978
##  [541] -10.2811688  -6.0594864 -14.6540896  -7.9722791  -7.6956017
##  [546] -14.0683362  -4.7781522 -14.3846045 -13.4066523 -14.2390964
##  [551] -14.7728983 -11.7514183  -8.4133920 -11.9015538  -8.5099234
##  [556] -14.8638379  -6.5523812 -14.0782748  -9.4189950  -4.6541283
##  [561] -17.4503927 -12.3188022  -6.8324568 -10.5711449  -9.4633615
##  [566] -10.6773123 -17.4855693  -5.6095180 -10.1052170  -9.5214371
##  [571] -16.6857188 -12.1546135  -8.3123856  -7.9942903  -3.2478963
##  [576] -14.3522175  -9.0164644 -12.9971932 -10.8776455  -8.9217443
##  [581] -15.1978472 -18.9551359  -6.2781919 -13.1936248  -6.3687983
##  [586]  -7.2267198 -10.0932211 -17.3489966  -7.6980444  -6.9747232
##  [591] -13.6405977 -13.0375382 -12.3630383  -7.7793461 -10.4766837
##  [596]  -7.0440332 -15.0294584  -6.8503930 -14.8739828 -10.6212715
##  [601] -11.9095645 -11.0079593  -9.8362763  -9.8402864  -5.8112301
##  [606] -11.3598963  -3.4563852 -12.6449565 -10.2964459  -4.3674439
##  [611]  -5.3214586 -10.2820226 -14.4612341 -15.3415452 -11.9957640
##  [616]  -6.8378797  -9.9594296 -11.8015683  -5.1311314  -9.0657716
##  [621]  -2.7813718 -11.1998897  -6.4664831 -10.3060134  -7.3264199
##  [626]  -9.8881630  -8.1508930  -9.2787762 -12.1705759 -11.4096312
##  [631]  -7.5578166 -15.9633568 -14.8345667 -11.2138887 -10.0571918
##  [636] -13.5737809  -8.0033992 -12.0130503  -5.7639919 -10.3058218
##  [641] -12.0518916  -7.9246277  -7.5683105  -9.5709656  -4.0637389
##  [646] -10.0216174 -10.2375519 -11.5193789 -11.6686663  -6.5572113
##  [651] -12.0921734 -17.7491367 -11.4723108  -7.7254396  -9.4372685
##  [656] -11.9150577  -6.3468112 -14.7027474 -10.6161027 -16.0671251
##  [661] -10.9797979 -12.4964656  -9.2110881 -10.0540978  -5.1839016
##  [666] -15.1996024 -10.9888214 -14.3079963 -10.4287906 -10.6269983
##  [671] -16.6444821 -10.5878586 -15.4910532  -9.5074162  -8.2292933
##  [676] -10.9420346 -10.9141148 -15.2850647  -7.5057845  -9.6817147
##  [681]  -8.9487458  -7.9655167 -10.0711337 -10.8619394  -9.3519818
##  [686] -15.0747446 -13.3488213  -9.5538598  -7.5000251 -15.4552399
##  [691] -13.4506885  -7.2361176 -13.4495349 -13.0575483  -9.7920270
##  [696] -11.3581630  -9.9814782  -9.3877104  -7.7359869 -11.3324096
##  [701] -13.7361363  -7.3287641 -10.7689014  -5.9931585  -6.0395483
##  [706]  -9.1132312  -5.7106684 -12.8085110 -12.6485650 -12.6706439
##  [711] -11.1410171  -6.7615414  -8.9664639  -8.3229848  -8.3194229
##  [716] -14.1252653 -14.8457194  -5.6020804  -4.0912776  -1.5210499
##  [721]  -7.8673433 -12.1965650 -15.2251476  -5.0210976  -9.9917918
##  [726] -15.2617071 -13.4179643  -6.9500670  -5.0190397  -8.4545446
##  [731]  -6.8325154 -10.1397893 -10.9479484  -7.2893254  -4.9308554
##  [736] -10.1918829 -10.9932357  -7.3574838 -13.0867630 -10.2305872
##  [741]  -5.0008261 -16.7143584  -9.7488854  -5.2236137  -8.3341065
##  [746] -13.3852944 -11.1981161  -7.5814518  -8.0339803 -10.1097160
##  [751] -10.9805808 -12.3867060 -10.9009620  -9.2482867 -11.4680546
##  [756] -14.0327933 -13.2870854 -14.8047308  -6.4718515 -12.3712712
##  [761] -15.3441786 -11.4058931 -12.6482450 -12.3589393 -11.0911152
##  [766]  -8.3889158  -3.2957022  -7.6550820  -3.7570248 -12.4761730
##  [771]  -8.2169081  -6.9151626 -12.8871098 -11.6990543 -10.0598020
##  [776]  -9.6708487  -9.6055945 -15.3285469  -8.1752903  -9.3847353
```

```
##  [781]  -2.6688777 -10.6666717  -8.0689355  -8.6899207  -9.1905077
##  [786] -10.1251848 -10.6528902 -10.4945548 -15.2871513 -11.7700994
##  [791] -10.0205520  -3.9363164  -9.9404014 -16.1388249 -10.6099891
##  [796]  -7.2501144 -10.1425939 -10.9246165  -9.6853010  -6.1362968
##  [801] -15.9212856  -8.6395003 -10.5348246 -11.7348319  -7.4614210
##  [806]  -6.9870315 -11.5486209 -11.8085799 -14.4214414  -8.1189880
##  [811]  -6.8115684 -10.2455970  -5.6807455  -3.3541264 -15.9291579
##  [816]  -3.1780146 -11.5834791 -16.8052110  -3.0245034 -12.1023325
##  [821] -11.4060510 -16.0818295  -5.0630911 -12.5244189  -9.0547434
##  [826]  -7.8015761 -13.3624925  -9.1500400  -8.4190270 -11.4086649
##  [831] -19.3367067  -4.4928651  -5.3833947 -13.6047675  -8.7705925
##  [836] -10.5470474  -6.4701910 -12.3350706  -5.7207740  -5.8911353
##  [841]  -8.7021226 -11.9978794  -9.5147660 -14.9733569 -14.7362723
##  [846] -12.7393846 -13.0010604 -12.8212835  -6.3625969 -11.5803708
##  [851] -11.8159022  -8.1294680  -4.3540929 -12.4788406  -8.3155656
##  [856] -12.6803537  -9.3358054  -9.7913918 -11.2470519  -6.4328442
##  [861] -14.6762639  -6.9334697  -7.6390064  -7.6590754 -14.0990093
##  [866]  -5.9297816  -8.9286403 -12.6169829 -13.8547305  -5.4620468
##  [871] -10.2460926 -15.0143735 -16.2265748 -10.5303377  -7.2286307
##  [876]  -9.7482953 -15.4693506 -13.0263615  -3.9481207 -12.8567027
##  [881] -11.5160853  -9.4685931 -12.0346223  -5.1855500  -7.2608957
##  [886] -10.6821118  -9.2156405  -5.5272081 -15.8269964 -12.8875012
##  [891] -13.4341866  -7.8582779  -8.6778148  -3.0989991  -6.2873467
##  [896]  -9.5682642  -8.7943788  -9.4977611  -7.6195386  -6.8838740
##  [901] -12.5588361 -10.0370242  -0.9557472  -8.8185876  -5.2630580
##  [906]  -9.3566076  -7.8358182 -14.3279679  -8.7940862  -3.2261721
##  [911] -12.0571242  -0.7710954 -13.1345390  -9.8397829  -3.6087606
##  [916]  -8.6228320 -13.6904252 -16.1622424 -10.1828968  -9.8862898
##  [921] -11.3851798  -8.7133466 -11.2358668  -8.7734400  -8.9407217
##  [926]  -5.7432043 -10.0405896 -12.8232333  -5.8468463  -9.5543047
##  [931] -11.5014291 -15.4368886  -9.1088830  -7.2657561 -12.6339018
##  [936]  -5.0469417  -5.4118069  -9.6434181 -14.4171346 -10.5533930
##  [941]  -8.2601420  -5.8828181 -12.7936096  -8.0305740 -15.1522772
##  [946] -11.9777622 -16.1240678  -8.4772523  -7.0357779 -12.4407519
##  [951] -11.5901686 -11.8816433 -11.8410468 -14.1003062  -7.9426030
##  [956]  -7.6669412  -6.8282525 -16.2201569  -9.4010924  -7.7932283
##  [961] -12.1080420 -15.7104850 -11.0720956 -13.3225164 -10.1964139
##  [966] -12.0659285  -9.4288907  -7.8960463  -8.2913135  -9.8402600
##  [971]  -9.7401660 -11.5680258 -15.1638862 -12.6122252 -10.2195339
##  [976]  -9.1963706  -7.4899589 -13.5838991 -12.5017219 -17.4737353
##  [981] -11.7184093  -7.7451850  -9.3783199  -9.8402192  -6.0371248
##  [986] -12.1759910 -13.2107903  -7.7835186  -9.5362645  -9.4295271
##  [991] -14.1284158  -8.0289523 -12.9959683  -9.3534887  -8.9695997
##  [996]  -8.1867848 -10.9539635 -12.6753714  -6.2143992  -7.6956776
```

```
#TO-DO
```

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

```
mean(v)
```

```
## [1] -10.0928
```

```
standard_error = sqrt(10)/sqrt(1000)
standard_error
```

```
## [1] 0.1
```

```
#TO-DO
```

Find the 5%ile of `v` and use the `qnorm` function as part of a test to ensure it is correct based on probability theory.

```
x = as.numeric(quantile(v, probs = .05))
x
```

```
## [1] -15.49159
```

```
y = as.numeric(qnorm(0.05, mean = -10, sd = sqrt(10), lower.tail = T, log.p = F))
y
```

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

```
expect_equal(x, y, .2)

#TO-DO
#expect_equal(..., tol = )
```

Find the sample quantile corresponding to the value -7000 of `v` and use the `pnorm` function as part of a test to ensure it is correct based on probability theory.

```
a = as.numeric(pnorm(0.05, mean = -10, sd = sqrt(10), lower.tail = T, log.p = F))
a
```

```
## [1] 0.9992588
```

```
inverse_quantile_v = ecdf(v[])
b = as.numeric(inverse_quantile_v(-7000))
b
```

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

```
expect_equal(a, b, 1)
#TO-DO
#expect_equal(..., tol = )
```

7. 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.

```
my_list = list()
my_list$a = "A"
my_list$b = "B"
my_list$c = "C"
my_list$d = "D"
my_list$e = "E"
my_list$f = "F"
my_list$g = "G"
my_list$h = "H"
value_a = array(1, 1)
value_b = array(1:4, dim = c(2, 2))
value_c = array(1:27, dim = c(3, 3, 3))
value_d = array(1:256, dim = c(4, 4, 4, 4))
value_e = array(1:3125, dim = c(5, 5, 5, 5, 5))
value_f = array(1:46656, dim = c(6, 6, 6, 6, 6, 6))
value_g = array(1:823543, dim = c(7, 7, 7, 7, 7, 7, 7))
value_h = array(1:16777216, dim = c(8, 8, 8, 8, 8, 8, 8, 8))
my_list[[ "A" ]] <- value_a
```

```
my_list[[ "B" ]] <- value_b
my_list[[ "C" ]] <- value_c
my_list[[ "D" ]] <- value_d
my_list[[ "E" ]] <- value_e
my_list[[ "F" ]] <- value_f
my_list[[ "G" ]] <- value_g
my_list[[ "H" ]] <- value_h

#I wasn't sure what the question was asking with making 8 Entries, or if we needed to make functions so



#TO-DO
```

Test with the following uncomprehensive tests:

{r} # 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
## 96 bytes
##
## $b
## 96 bytes
##
## $c
## 96 bytes
##
## $d
## 96 bytes
##
## $e
## 96 bytes
##
## $f
## 96 bytes
##
## $g
## 96 bytes
##
## $h
## 96 bytes
##
## $A
## 208 bytes
##
## $B
## 216 bytes
##
```

```
## $C
## 336 bytes
##
## $D
## 1232 bytes
##
## $E
## 12728 bytes
##
## $F
## 186848 bytes
##
## $G
## 3294400 bytes
##
## $H
## 67109088 bytes
```

Use `?lapply` and `?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.

So its giving us the the size of each key within our list. If you notice the first key%a it's just 96 bytes because its only holding the character "A" but key$A has 208 repesenting the array of numbers it holds, and you'll notice the size of the keys are large for those with higher dimensions hence why H gives you the largest size.

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

```
rm(list = ls())
#TO-DO
```

Basic Binary Classification Modeling

8. 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.

```
data("iris")
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

```
#TO-DO
```

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.

```
iris = iris[iris$Species != "virginica",]
#rm(virginica, list = character(), pos = -1, inherits = F)
#TO-DO
#QUICKNOTE In all honesty I got a lot of help from friends on this question and onward, most of the wor
```

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".

```
nrow(iris[iris$Species=="versicolor",])
```

```
## [1] 50
```

```
y = c(rep(0,nrow(iris[iris$Species=="setosa",])),
      rep(1,nrow(iris[iris$Species=="versicolor",])))
#TO-DO
```

9. 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?

```
X = data.frame(iris$Sepal.Length)
z = cbind(X, y)
n = nrow(z)

max = 1000
k = c(0, 0)

for (j in 1 : max){
  for (i in 1 : n){
    f = c(1, z[i, 1])
    h = ifelse(sum(f * k) > 0, 1, 0)
    k = k + (y[i] - h) * f
  }
}
k
```

```
## [1] -143.0   33.7
```

```
#TO-DO
```

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.
##
```

Write your answer here in English.

Well the Mean is 5.006 very close to the Median so it's basically 50/50 between setosa and virginica thus it make sense.

10. Fit a perceptron model explaining `y` using all three features. Try to write your own code to do this. Provide the estimated parameters (i.e. the four entries of the weight vector)? What is the total number of errors this model makes?

```
X = data.frame(iris)
X$Species = NULL
q = c(rep(0,100))
z = cbind(X, y, q)

n = nrow(z)
k = c(0, 0, 0, 0)
max = 1000
error = 0
error_count = 0

for (j in 1 : max){
  for (i in 1 : n){
    f = c(z[i,1], z[i,2], z[i,3], z[i,4])
    h = ifelse(sum(f * k) > 0, 1, 0)
    q[i] = ifelse((h != y[i]), 1, 0)
    k = k + (z[i,5] - h) * f
  }
}
k
```

```
## [1] -1.1 -3.6  5.2  2.2
```

```
total_error = sum(1==z[,"q"])
error_rate = total_error / nrow(z)
error_rate
```

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

```
#TO-DO
```