Lab 1

Amir ElTabakh

11:59PM February 18, 2021

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 to your repository in a directory called "labs".

- Print out the numerical constant pi with ten digits after the decimal point using the internal constant `pi`.

```
options(digits=11)
x <- pi
x
```

```
## [1] 3.1415926536
```

- Sum up the first 103 terms of the series 1 + 1/2 + 1/4 + 1/8 + …

```
sum(1/(2^(0:102)))
```

```
## [1] 2
```

- Find the product of the first 37 terms in the sequence 1/3, 1/6, 1/9 …

```
prod(1/(3*(1:37)))
```

```
## [1] 1.613528728e-61
```

```
prod(1/seq(from=3, by=3, length.out=37))
```

```
## [1] 1.613528728e-61
```

- Find the product of the first 387 terms of `1 * 1/2 * 1/4 * 1/8 *` …

```
prod(1/(2^(0:386)))
```

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

Is this answer *exactly* correct?

This answer is not exactly correct, the program is rounding to zero.

- Figure out a means to express the answer more exactly. Not compute exactly, but express more exactly.

```
sum(log(1/(2^(0:386))))
```

```
## [1] -51771.856063
```

```
-log(2)*sum(0:386)
```

```
## [1] -51771.856063
```

- Create the sequence `x = [Inf, 20, 18, ..., -20]`.

```
x <- c(Inf, seq(from=20, to=-20, by=-2))
x
```

```
##  [1] Inf  20  18  16  14  12  10   8   6   4   2   0  -2  -4  -6  -8 -10 -12 -14
## [20] -16 -18 -20
```

Create the sequence `x = [log_3(Inf), log_3(100), log_3(98), ... log_3(-20)]`.

```
x <- c(Inf, seq(from=100, to=-20, by=-2))
x <- log(x, base=3)
```

```
## Warning: NaNs produced
```

```
log(100, 3)
```

```
## [1] 4.1918065486
```

Comment on the appropriateness of the non-numeric values.

NAN occurs because you cannot take the log of a negative number. -Inf occurs when you take the log of 0.

- Create a vector of booleans where the entry is true if `x[i]` is positive and finite.

```
y = !is.nan(x) & is.finite(x) & x > 0
y
```

```
##  [1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [25]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [37]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [49]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [61] FALSE FALSE
```

- Locate the indices of the non-real numbers in this vector. Hint: use the `which` function. Don't hesitate to use the documentation via `?which`.

```
?which
```

```
## starting httpd help server ... done
```

```
which(!y)
```

```
##  [1]  1 52 53 54 55 56 57 58 59 60 61 62
```

```
which(y == FALSE)
```

```
##  [1]  1 52 53 54 55 56 57 58 59 60 61 62
```

- Locate the indices of the infinite quantities in this vector.

```
which(is.infinite(x))
```

```
## [1]  1 52
```

- Locate the indices of the min and max in this vector. Hint: use the `which.min` and `which.max` functions.

```
which.min(x)
```

```
## [1] 52
```

```
which.max(x)
```

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

- Count the number of unique values in `x`.

```
length(unique(x))
```

```
## [1] 53
```

- Cast `x` to a factor. Do the number of levels make sense?

```
as.factor(x)
```

```
##  [1] Inf               4.19180654857877  4.1734172518943   4.15464876785729
##  [5] 4.13548512895119  4.11590933734319  4.09590327428938  4.07544759935851
##  [9] 4.05452163806914  4.03310325630434  4.01116871959141  3.98869253500376
## [13] 3.96564727304425  3.94200336638929  3.91772888178973  3.89278926071437
## [17] 3.86714702345081  3.84076143030548  3.81358809221559  3.78557852142874
## [21] 3.75667961082847  3.72683302786084  3.69597450568212  3.66403300987579
## [25] 3.63092975357146  3.59657702661571  3.56087679500731  3.52371901428583
## [29] 3.48497958377173  3.44451784578705  3.40217350273288  3.3577627814323
## [33] 3.31107361281783  3.26185950714291  3.20983167673402  3.15464876785729
## [37] 3.09590327428938  3.03310325630434  2.96564727304425  2.89278926071437
## [41] 2.8135880922156   2.72683302786084  2.63092975357146  2.52371901428583
## [45] 2.40217350273288  2.26185950714291  2.09590327428938  1.89278926071437
## [49] 1.63092975357146  1.26185950714291  0.630929753571457 -Inf
## [53] NaN               NaN               NaN               NaN
## [57] NaN               NaN               NaN               NaN
## [61] NaN               NaN
## 53 Levels: -Inf 0.630929753571457 1.26185950714291 ... NaN
```

- Cast x to integers. What do we learn about R's infinity representation in the integer data type?

```
as.integer(x)
```

```
## Warning: NAs introduced by coercion to integer range
```

```
##  [1] NA  4  4  4  4  4  4  4  4  4  4  3  3  3  3  3  3  3  3  3  3  3  3  3  3
## [26]  3  3  3  3  3  3  3  3  3  3  3  3  3  2  2  2  2  2  2  2  2  2  1  1  1
## [51]  0 NA NA NA NA NA NA NA NA NA NA NA
```

- Use x to create a new vector y containing only the real numbers in x.

```
y = x[!is.nan(x) & is.finite(x)]
y
```

```
##  [1] 4.19180654858 4.17341725189 4.15464876786 4.13548512895 4.11590933734
##  [6] 4.09590327429 4.07544759936 4.05452163807 4.03310325630 4.01116871959
## [11] 3.98869253500 3.96564727304 3.94200336639 3.91772888179 3.89278926071
## [16] 3.86714702345 3.84076143031 3.81358809222 3.78557852143 3.75667961083
## [21] 3.72683302786 3.69597450568 3.66403300988 3.63092975357 3.59657702662
## [26] 3.56087679501 3.52371901429 3.48497958377 3.44451784579 3.40217350273
## [31] 3.35776278143 3.31107361282 3.26185950714 3.20983167673 3.15464876786
## [36] 3.09590327429 3.03310325630 2.96564727304 2.89278926071 2.81358809222
## [41] 2.72683302786 2.63092975357 2.52371901429 2.40217350273 2.26185950714
## [46] 2.09590327429 1.89278926071 1.63092975357 1.26185950714 0.63092975357
```

- Use the left rectangle method to numerically integrate x^2 from 0 to 1 with rectangle width size 1e-6.

```
sum(seq(from=0, to=1-(1e-6), by=1e-6)^2)*1e-6
```

```
## [1] 0.33333283333
```

- Calculate the average of 100 realizations of standard Bernoullis in one line using the `sample` function.

```
sum(sample(c(0,1), size=100, replace=TRUE))/100
```

```
## [1] 0.49
```

- Calculate the average of 500 realizations of Bernoullis with p = 0.9 in one line using the `sample` and `mean` functions.

```
sum(sample(c(0,1), size=500, replace=TRUE, prob=c(0.1, 0.9)))/500
```

```
## [1] 0.906
```

- Calculate the average of 1000 realizations of Bernoullis with p = 0.9 in one line using `rbinom`.

```
?rbinom
rbinom(n=1000, size=1, p=0.9)
```

```
##    [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1
##   [38] 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 1 1 1 1 1 1 0
##   [75] 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1
##  [112] 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 0 1 1 1
##  [149] 1 0 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 1 1 1 1 1
##  [186] 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 1 1 1 1 1 1 1
##  [223] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
##  [260] 1 1 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1
##  [297] 1 1 0 1 1 1 1 0 1 1 1 1 1 0 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1
##  [334] 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 0 1 1 0 1 1 1 1 1 1 1 1
##  [371] 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1
##  [408] 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 1 1 0 1 1 0 1
##  [445] 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
##  [482] 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 0 1 0 1 0 1 1
##  [519] 1 0 1 1 1 0 0 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
##  [556] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [593] 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1
##  [630] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [667] 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 1 1
##  [704] 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 1 1 0 1 0 1 0 1 1 1 1 1
##  [741] 1 1 1 0 0 0 1 1 0 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
##  [778] 1 0 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [815] 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 0 1 1 1
##  [852] 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
##  [889] 1 0 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 0 1 1 1 1 1 1 1
##  [926] 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1
##  [963] 0 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1000] 1
```

- In class we considered a variable `x_3` which measured "criminality". We imagined L = 4 levels "none", "infraction", "misdimeanor" and "felony". Create a variable `x_3` here with 100 random elements (equally probable). Create it as a nominal (i.e. unordered) factor.

```
x_3 = as.factor(sample(c("none",  "infraction", "misdimeanor", "felony"), size=100, replace=TRUE))
x_3
```

```
##   [1] infraction  infraction  none        felony      misdimeanor misdimeanor
##   [7] felony      misdimeanor misdimeanor felony      misdimeanor infraction 
##  [13] infraction  none        infraction  misdimeanor infraction  misdimeanor
##  [19] felony      none        infraction  none        none        infraction 
##  [25] felony      infraction  none        none        misdimeanor infraction 
##  [31] misdimeanor felony      felony      felony      felony      none
##  [37] felony      infraction  felony      none        none        none       
##  [43] none        misdimeanor none        none        infraction  infraction 
##  [49] felony      felony      felony      infraction  felony      infraction 
##  [55] misdimeanor infraction  felony      felony      felony      felony     
##  [61] felony      felony      infraction  felony      misdimeanor felony     
##  [67] infraction  misdimeanor infraction  none        misdimeanor misdimeanor
##  [73] infraction  infraction  infraction  infraction  none        felony     
##  [79] misdimeanor felony      infraction  felony      felony      felony     
##  [85] none        infraction  misdimeanor infraction  misdimeanor felony     
##  [91] felony      misdimeanor misdimeanor none        misdimeanor infraction 
##  [97] misdimeanor felony      felony      none       
## Levels: felony infraction misdimeanor none
```

- Use `x_3` to create `x_3_bin`, a binary feature where 0 is no crime and 1 is any crime.

```
x_3_bin = x_3 != "none"
x_3_bin
```

```
##   [1]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [13]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE
##  [25]  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
##  [37]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE
##  [49]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [61]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
##  [73]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [85] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
##  [97]  TRUE  TRUE  TRUE FALSE
```

- Use `x_3` to create `x_3_ord`, an ordered factor variable. Ensure the proper ordinal ordering.

```
x_3_ord = factor(x_3, levels = c("none",  "infraction", "misdimeanor", "felony"), order=TRUE)
x_3_ord
```

```
##   [1] infraction  infraction  none        felony      misdimeanor misdimeanor
##   [7] felony      misdimeanor misdimeanor felony      misdimeanor infraction 
##  [13] infraction  none        infraction  misdimeanor infraction  misdimeanor
##  [19] felony      none        infraction  none        none        infraction 
##  [25] felony      infraction  none        none        misdimeanor infraction 
##  [31] misdimeanor felony      felony      felony      felony      none       
##  [37] felony      infraction  felony      none        none        none       
##  [43] none        misdimeanor none        none        infraction  infraction 
##  [49] felony      felony      felony      infraction  felony      infraction 
##  [55] misdimeanor infraction  felony      felony      felony      felony     
##  [61] felony      felony      infraction  felony      misdimeanor felony     
##  [67] infraction  misdimeanor infraction  none        misdimeanor misdimeanor
##  [73] infraction  infraction  infraction  infraction  none        felony     
##  [79] misdimeanor felony      infraction  felony      felony      felony     
##  [85] none        infraction  misdimeanor infraction  misdimeanor felony     
##  [91] felony      misdimeanor misdimeanor none        misdimeanor infraction 
##  [97] misdimeanor felony      felony      none       
## Levels: none < infraction < misdimeanor < felony
```

- Convert this variable into three binary variables without any information loss and put them into a data matrix.

```
x_3_matrix = matrix(nrow = length(x_3), ncol = 3)
x_3_matrix[ ,1] = as.numeric(x_3 == "infraction")
x_3_matrix[ ,2] = as.numeric(x_3 == "felony")
x_3_matrix[ ,3] = as.numeric(x_3 == "misdimeanor")
colnames(x_3_matrix) = c("infraction", "felony", "is_misdimeanor")
x_3_matrix
```

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

- What should the sum of each row be (in English)?

The sum of each row should be 1 or 0. If the individual has a record of 'none', that will be captured by a row sum of zero.

Verify that.

```
rowSums(x_3_matrix)
```

```
##   [1] 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 1 1 0 0 1 1 1 1 1 1 1 0 1
##  [38] 1 1 0 0 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1
##  [75] 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0
```

- How should the column sum look (in English)?

We should expect for there to be about 25 values per column. This is assuming the sample() function uniformly distributes values.

Verify that.

```
colSums(x_3_matrix)
```

```
##     infraction         felony is_misdimeanor 
##             27             32             22
```

- Generate a matrix with 100 rows where the first column is realization from a normal with mean 17 and variance 38, the second column is uniform between -10 and 10, the third column is poisson with mean 6, the fourth column in exponential with lambda of 9, the fifth column is binomial with n = 20 and p = 0.12 and the sixth column is a binary variable with exactly 24% 1's dispersed randomly. Name the rows the entries of the `fake_first_names` vector.

```
fake_first_names = c(
  "Sophia", "Emma", "Olivia", "Ava", "Mia", "Isabella", "Riley",
  "Aria", "Zoe", "Charlotte", "Lily", "Layla", "Amelia", "Emily",
  "Madelyn", "Aubrey", "Adalyn", "Madison", "Chloe", "Harper",
  "Abigail", "Aaliyah", "Avery", "Evelyn", "Kaylee", "Ella", "Ellie",
  "Scarlett", "Arianna", "Hailey", "Nora", "Addison", "Brooklyn",
  "Hannah", "Mila", "Leah", "Elizabeth", "Sarah", "Eliana", "Mackenzie",
  "Peyton", "Maria", "Grace", "Adeline", "Elena", "Anna", "Victoria",
  "Camilla", "Lillian", "Natalie", "Jackson", "Aiden", "Lucas",
  "Liam", "Noah", "Ethan", "Mason", "Caden", "Oliver", "Elijah",
  "Grayson", "Jacob", "Michael", "Benjamin", "Carter", "James",
  "Jayden", "Logan", "Alexander", "Caleb", "Ryan", "Luke", "Daniel",
  "Jack", "William", "Owen", "Gabriel", "Matthew", "Connor", "Jayce",
  "Isaac", "Sebastian", "Henry", "Muhammad", "Cameron", "Wyatt",
  "Dylan", "Nathan", "Nicholas", "Julian", "Eli", "Levi", "Isaiah",
  "Landon", "David", "Christian", "Andrew", "Brayden", "John",
  "Lincoln"
)

n <- 100
X <- matrix(nrow=n, ncol=6)
X[,1] <- rnorm(n=n, mean=17, sd=sqrt(38))
X[,2] <- runif(n=n, min=-10, max=10)
X[,3] <- rpois(n=n, lambda=6)
X[,4] <- rexp(n=n, rate=9)
X[,5] <- rbinom(n=n, size=20, p=0.12)
X[,6] <- sample(c(rep(1, n * 0.24), rep(0, n*0.76)))

rownames(X) = fake_first_names

X
```

```
##                     [,1]           [,2] [,3]            [,4] [,5] [,6]
## Sophia    19.5124363257  6.13117717672    6 0.1140487433054    2    0
## Emma      17.8075901170  1.49189714342   11 0.4142925977060    3    0
## Olivia    11.3901089510 -0.27938747779    5 0.0790967831033    1    0
## Ava       13.2717423148  8.07859887835    8 0.0522556416690    6    0
## Mia       17.4199813812 -5.39675085805   11 0.1147867872116    2    1
## Isabella  23.2590398226 -6.88128389884    1 0.2521878727054    1    0
## Riley     14.0298719082  5.04345212132    9 0.0130567526859    2    0
## Aria      21.5323058697 -8.83130562026    2 0.0895410828965    2    0
## Zoe       22.0891534522  3.89889195561    6 0.0305771632068    0    0
## Charlotte 23.2071222215 -5.66520791966    7 0.0166097955985    2    0
## Lily      15.6108549089 -4.47846951894    6 0.1511111503627    2    0
## Layla     20.1100214190  2.42999473121    3 0.0727419548543    4    1
## Amelia    13.9866340003 -4.79999087751    7 0.0475734700449    3    0
## Emily     28.2578092937 -8.82616977673    3 0.3333310137006    3    0
## Madelyn   13.5071740799 -1.43941889983    2 0.1203042264567    1    1
## Aubrey    17.5279857233 -0.75556893833    7 0.1602992274707    2    0
## Adalyn    13.9738794653 -6.58720296342    7 0.0080225320222    4    0
## Madison   17.3334257378 -2.04695395660    5 0.0372743220586    3    1
## Chloe     12.6347312503 -5.98141315393    4 0.0115718082525    2    1
## Harper    21.7963158623  6.56758550555    9 0.0022073805760    1    0
## Abigail   20.7631051118 -9.23048926983    6 0.4003449634726    1    0
## Aaliyah   21.3634184407 -6.84209550265    5 0.0893037838244    4    0
## Avery      7.6058259437 -8.68789581582    7 0.2050691513872    3    0
## Evelyn    31.3043707496  8.04461730644    4 0.0552793707945    3    0
## Kaylee    13.6939169101  9.06808945816    5 0.0955045498615    2    0
## Ella      10.0070220329  9.43226655480    4 0.0539741123923    4    0
## Ellie     22.4512827445  0.48931350000    2 0.0775130817252    2    0
## Scarlett  19.9846200389 -5.11447096709   10 0.0373493080222    5    0
## Arianna   20.3364897776 -1.52604652103    7 0.0215037596707    2    0
## Hailey    20.9064953136  9.88538612146   11 0.1016160594253    2    0
## Nora      19.6511141846  0.75326569844    7 0.2842924013180    2    1
## Addison   16.4623863432  0.25087298825    7 0.2419908160411    4    0
## Brooklyn  15.1203830541  0.59217149392    7 0.0065883383924    0    1
## Hannah    24.3104723020 -4.32599717751    6 0.0662820151386    2    1
## Mila      19.2948400186 -8.69583406486    9 0.1918576502458    3    0
## Leah       9.9069516055  7.55635380745    4 0.0118784954394    3    0
## Elizabeth 25.8832093998  4.16241603903   10 0.0518002659600    0    1
## Sarah     14.0877925190 -5.26327606291    7 0.0886178915860    1    0
## Eliana    15.0032716492  3.20167207159    7 0.2602331329825    3    1
## Mackenzie 31.0342947035  6.83708556928    7 0.0639826646592    0    0
## Peyton    21.7443955360 -5.83290692419    7 0.5635006611633    2    1
## Maria     15.0364379061  3.91389633995    6 0.1012597161297    4    0
## Grace     15.9586009522  2.82240050379    3 0.0626873319141    2    0
## Adeline   15.7440033289 -6.06800736859    7 0.0035722515980    0    0
## Elena     21.9517108501 -1.05980598368    8 0.1665131420632    4    0
## Anna      20.9303320002  8.13079005107    5 0.0201014762537    3    0
## Victoria  20.9068585381  1.11173335928   11 0.1992957498785    3    1
## Camilla   12.0340015442  5.99554708228    5 0.2571210751429    2    0
## Lillian   20.9434027669 -9.71302066930   10 0.0497770366362    4    1
## Natalie   13.8354908316  2.39739262965    6 0.0330395437777    3    0
## Jackson   10.7842000527  9.14642752614    6 0.0893381306540    6    0
## Aiden     13.6008975259  7.62514570262    3 0.0400270528367    2    0
## Lucas     21.7052012922 -7.03677630983    2 0.3563811704330    2    0
## Liam      11.8541585999  8.60007264186    7 0.1432845728011    3    0
## Noah      14.0122828453 -3.30034088809    5 0.0365837006830    4    1
## Ethan     14.2444503489 -5.06105064880    8 0.2557098000249    1    0
## Mason     20.9456483358  1.87065470032    9 0.0264643155970    2    0
## Caden     27.5463138824  3.88727381825    6 0.0164239277753    2    1
## Oliver    23.0209464193 -0.85328714456    4 0.1186918123682    4    1
## Elijah    17.7191720366 -9.08202291001    5 0.2028691669212    0    0
## Grayson   15.2008222757 -0.98467234522    2 0.0190709827173    2    0
## Jacob     11.9804641373 -5.28235884849    2 0.3000439543965    1    0
## Michael   33.1963658564 -4.79753760155    7 0.0027126834935    2    0
## Benjamin  20.5169877733 -7.92998590507    6 0.0943099179261    4    1
## Carter    14.7103383869  7.95302613638   10 0.0167892796826    6    0
## James     23.2863851063 -3.08093557134    4 0.3665782461132    1    0
## Jayden    16.2040730981 -6.06894807424    9 0.0967622700932    2    0
## Logan     18.0061981271  1.35660210624    5 0.1730856271800    0    0
## Alexander 11.9518634924  6.59925982822    3 0.0780431398663    3    0
## Caleb      4.6154575181  7.17953393236    3 0.5195346660632    3    0
## Ryan      15.6602872540 -0.78060525004    3 0.0088210632439    2    1
## Luke      28.5840337658 -6.70666369610    4 0.2603312262456    3    0
## Daniel     9.9800403115  5.57805249002    4 0.0686386326431    3    0
## Jack      10.2234659423 -7.31879010331    7 0.1462484038285    6    0
## William   12.7024982431 -0.94297029078    4 0.3882035552808    3    0
## Owen      19.2745567692 -3.04749839939    8 0.0744807267975    3    1
## Gabriel   13.4415932329  4.65159135871    8 0.1020928387443    4    1
## Matthew   20.0816232546  2.06495203543    7 0.1749245499472    3    1
## Connor    17.2875217687 -9.30518577807    8 0.0426802340791    4    0
## Jayce     15.0646931328 -7.94796291273    6 0.0172347796357    4    0
## Isaac     14.4449691393 -4.45233939216    3 0.1824346979366    3    0
## Sebastian 16.6037446445  0.51526531111    8 0.2926098801470    4    1
## Henry     21.8986378516 -7.15383071452    5 0.3481346807359    2    0
## Muhammad  21.9166261133 -6.11098980065    6 0.4115452077006    3    0
## Cameron   17.1453207750  4.25595008302    8 0.1117404465852    3    0
## Wyatt     16.0753509506 -5.34682201687    4 0.1237013474474    1    0
## Dylan     15.4766781925  7.71895005833    2 0.2701691146800    3    0
## Nathan    12.5589568454 -4.56151650287    7 0.0580824149462    2    0
## Nicholas  13.2671797161  0.55223745760    3 0.0013347465752    5    0
## Julian    14.5271680646 -3.41087585781    5 0.0897851246880    2    1
## Eli       22.3029370445  5.54512503557    2 0.2034026361495    3    0
## Levi      14.6722246699 -9.19034911320    6 0.2653968303014    3    0
## Isaiah    17.3102415852  6.84683867265    6 0.0393047206518    1    0
## Landon    18.3050038782  4.25256739371   11 0.0519811379620    3    0
## David     22.0212074327  6.57683383208    6 0.0565847251564    3    0
## Christian  7.1907608105  9.87610584591    8 0.1940685086273    0    0
## Andrew    19.0539968284  0.38290294819   13 0.0661023575813    3    0
## Brayden   10.7874257613  6.33624847047   10 0.0308911783595    4    0
## John      14.6470683093 -1.18393473793    2 0.0708758009908    1    1
## Lincoln   16.5253565894  7.07842460368   10 0.0158540134187    3    0
```

- Create a data frame of the same data as above except make the binary variable a factor “DOMESTIC” vs “FOREIGN” for 0 and 1 respectively. Use RStudio’s `View` function to ensure this worked as desired.

```
df = data.frame(X)
df$X6 = factor(df$X6, levels = c(0, 1), labels = c("DOMESTIC", "FOREIGN"))
View(df, "Lab 1 DF")
```

- Print out a table of the binary variable. Then print out the proportions of “DOMESTIC” vs “FOREIGN”.

```
table(df$X6)
```

```
##
## DOMESTIC  FOREIGN
##       76       24
```

```
table(df$X6)/100
```

```
##
## DOMESTIC  FOREIGN
##     0.76     0.24
```

Print out a summary of the whole dataframe.

```
summary(df)
```

```
##        X1                    X2                     X3
##  Min.   : 4.6154575   Min.   :-9.71302067   Min.   : 1.00
##  1st Qu.:13.9834454   1st Qu.:-5.35930423   1st Qu.: 4.00
##  Median :16.8745327   Median :-0.51747821   Median : 6.00
##  Mean   :17.5064808   Mean   :-0.20530311   Mean   : 6.09
##  3rd Qu.:20.9439642   3rd Qu.: 5.16887035   3rd Qu.: 8.00
##  Max.   :33.1963659   Max.   : 9.88538612   Max.   :13.00
##        X4                      X5               X6
##  Min.   :0.0013347466   Min.   :0.00   DOMESTIC:76
##  1st Qu.:0.0388158675   1st Qu.:2.00   FOREIGN :24
##  Median :0.0893209572   Median :3.00
##  Mean   :0.1310310010   Mean   :2.56
##  3rd Qu.:0.1953753189   3rd Qu.:3.00
##  Max.   :0.5635006612   Max.   :6.00
```

- Let `n = 50`. Create a n x n matrix `R` of exactly 50% entries 0's, 25% 1's 25% 2's. These values should be in random locations.

```
n <- 50
R <- matrix(nrow=n, ncol=n, sample(c(rep(0, n*n*0.5), rep(1, n*n*0.25), rep(2, n*n*0.25))))
df <- data.frame(R)
df
```

```
##    X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20 X21
## 1   2  2  0  2  0  1  0  1  1   2   1   0   2   1   0   1   0   0   0   1   0
## 2   0  1  2  0  0  0  2  2  0   0   1   1   0   1   2   1   1   0   0   2   2
## 3   0  1  0  0  2  1  0  0  1   0   2   0   2   0   0   0   0   0   0   0   1
## 4   1  2  2  0  0  2  0  2  0   0   2   1   2   1   1   1   2   1   0   1   2
## 5   2  2  0  0  1  0  0  1  2   1   2   0   0   2   0   0   2   0   2   1   2
## 6   1  0  0  2  0  2  1  2  2   1   2   1   1   0   0   2   0   0   0   0   0
## 7   0  2  0  0  2  1  1  1  1   0   1   0   0   1   0   2   2   0   0   0   0
## 8   0  0  2  2  1  0  0  2  2   2   0   1   0   0   0   0   2   2   0   0   0
## 9   1  1  2  0  0  1  2  1  0   0   2   0   2   1   0   2   1   1   2   1   2
## 10  0  1  0  2  0  0  1  0  2   1   1   2   0   2   0   2   2   1   1   0   0
## 11  0  1  0  1  1  0  1  0  1   2   0   1   0   2   0   0   2   2   0   0   2
## 12  0  1  1  0  1  0  0  1  0   1   0   0   0   0   1   0   0   0   2   1   1
## 13  0  0  2  1  2  0  0  0  0   1   1   1   0   0   1   0   1   1   0   2   2
## 14  0  2  0  2  0  1  2  2  0   1   2   2   1   0   0   0   2   0   2   2   2
## 15  0  0  0  0  1  1  0  2  0   1   0   2   0   2   1   2   2   1   0   0   2
## 16  2  1  0  2  2  2  0  2  1   2   2   0   1   1   0   1   0   0   0   2   0
## 17  2  1  0  2  1  0  0  2  0   2   1   0   0   2   2   2   1   0   0   1   2
## 18  0  1  0  2  1  0  0  2  1   2   0   1   0   2   0   1   1   0   0   0   0
## 19  0  0  2  0  1  2  0  0  2   1   1   0   2   0   0   0   0   2   0   0   0
## 20  0  1  0  2  0  1  1  0  2   0   2   0   2   0   0   2   0   0   0   0   1
## 21  0  0  0  0  1  1  0  0  1   1   0   2   0   2   0   0   1   1   1   0   0
## 22  2  0  0  0  0  1  2  2  0   0   0   1   0   0   0   1   0   0   1   2   0
## 23  1  0  1  0  2  0  0  1  0   0   0   0   2   1   2   1   0   1   0   2   1
## 24  1  0  1  0  0  1  0  0  0   0   0   2   2   2   2   0   1   2   2   0   0
## 25  0  1  2  0  1  0  1  2  0   0   0   0   2   2   2   0   2   1   0   0   0
## 26  0  0  2  2  0  0  0  2  0   1   0   2   0   0   0   1   2   1   1   0   1
## 27  0  1  0  0  0  2  1  2  1   1   0   0   2   1   1   2   0   2   1   2   0
## 28  0  0  1  2  0  0  2  0  1   0   1   0   0   2   0   1   1   2   0   0   1
## 29  2  0  0  2  2  0  0  0  1   1   0   0   1   0   0   1   1   0   2   1   1
## 30  0  2  1  2  2  0  2  1  2   0   1   0   1   1   0   1   0   0   2   1   0
## 31  0  2  1  2  0  2  2  0  2   0   1   0   2   2   0   0   2   0   0   0   0
## 32  0  0  0  0  0  1  0  1  0   0   1   0   1   2   1   1   1   2   1   2   0
## 33  1  1  1  0  2  1  0  0  0   0   0   1   2   0   0   1   1   0   0   0   0
## 34  1  0  0  1  0  2  0  0  0   0   0   2   0   2   0   0   2   0   0   0   1
## 35  1  2  1  2  2  1  0  2  0   1   0   2   1   2   1   2   0   0   0   0   1
## 36  2  1  1  2  0  0  0  0  0   0   2   0   0   2   0   2   1   1   0   0   2
## 37  0  0  0  2  1  0  2  2  0   0   0   0   2   2   0   1   2   1   2   1   1
## 38  1  0  0  0  0  0  0  2  1   1   1   0   1   1   0   0   0   1   1   1   0
## 39  2  0  2  0  0  0  1  1  1   0   1   0   0   0   0   0   0   1   1   0   0
## 40  0  2  2  0  2  0  0  0  0   0   1   2   0   2   0   0   1   1   2   2   0
## 41  2  0  2  2  2  0  0  0  0   0   1   2   0   0   2   2   0   0   0   1   0
## 42  0  0  0  0  0  0  0  0  1   1   1   1   1   0   0   2   0   0   0   0   0
## 43  1  2  0  1  0  0  1  2  0   1   0   1   2   0   0   2   0   1   0   0   2
## 44  2  0  0  2  0  0  0  0  1   1   2   1   0   0   0   0   2   2   2   1   0
## 45  0  2  1  2  2  0  2  2  2   2   0   0   0   1   1   2   2   1   1   0   0
## 46  2  1  0  1  0  0  1  0  0   2   2   2   1   0   1   1   2   0   2   2   0
## 47  0  0  1  0  1  1  0  0  0   0   1   0   1   2   2   2   0   2   1   0   0
## 48  0  1  0  1  2  0  0  0  1   0   2   0   0   2   1   1   0   0   1   0   0
## 49  0  1  1  1  0  1  1  2  2   0   0   0   0   0   0   2   0   1   1   0   1
## 50  2  1  0  1  0  2  0  0  1   1   1   1   2   1   0   2   0   2   0   0   0
##    X22 X23 X24 X25 X26 X27 X28 X29 X30 X31 X32 X33 X34 X35 X36 X37 X38 X39 X40
## 1    0   0   0   0   2   0   2   2   1   1   2   0   0   1   0   1   0   0   0
## 2    1   2   2   1   2   0   0   2   0   0   1   0   2   0   0   0   0   0   0
## 3    2   0   0   1   0   2   1   0   0   0   1   1   0   1   1   1   1   2   1
## 4    0   2   2   2   1   0   2   2   2   0   2   0   0   0   2   0   0   0   2
## 5    1   1   1   2   2   0   2   1   0   1   0   0   0   0   0   0   0   0   0
## 6    1   0   2   0   2   1   0   2   0   2   1   0   2   2   0   1   0   0   2
## 7    0   1   1   1   1   2   0   2   2   0   1   0   0   2   0   1   1   1   2
## 8    0   0   2   0   0   1   2   0   0   0   0   2   0   2   1   0   0   0   0
## 9    2   0   2   1   0   1   1   2   0   1   0   1   0   2   2   1   2   0   0
## 10   0   0   0   0   0   0   2   2   2   0   1   0   0   0   1   0   0   0   2
## 11   0   2   0   2   2   0   1   2   0   0   0   2   0   1   0   1   0   2   1
## 12   2   2   2   0   1   1   1   0   2   2   0   0   2   0   0   0   0   1   1
## 13   1   0   0   2   1   2   0   1   1   2   0   0   2   0   2   0   0   2   0
## 14   0   2   0   2   0   1   1   1   0   1   1   0   0   0   0   1   1   0   2
## 15   1   2   0   1   1   2   0   0   2   2   1   1   0   0   2   2   0   1   0
## 16   0   0   0   2   0   0   1   2   2   0   0   1   1   1   0   0   0   1   0
## 17   0   1   2   0   0   2   0   1   1   0   0   2   0   0   0   0   0   0   0
## 18   0   0   2   2   2   0   0   1   2   0   2   1   2   2   2   0   1   2   2
## 19   0   2   0   0   0   0   1   1   0   2   0   0   0   0   0   2   1   2   1
## 20   2   0   2   1   0   1   1   0   2   1   0   0   0   0   0   0   2   0   1
## 21   0   0   0   2   0   0   0   1   2   2   2   2   0   2   0   1   0   0   0
## 22   0   2   1   0   0   2   1   2   1   1   2   1   0   1   0   0   2   2   1
## 23   0   0   0   1   1   0   0   0   1   2   2   2   0   1   0   0   2   0   0
## 24   0   2   2   1   2   1   2   0   1   2   2   1   0   0   0   1   0   1   0
## 25   1   0   0   2   1   2   1   0   0   0   2   0   0   1   0   0   2   2   0
## 26   2   0   0   2   0   1   1   0   0   0   0   0   1   1   1   2   0   0   2
## 27   1   0   0   0   0   2   1   1   1   0   0   2   2   0   2   1   0   0   0
## 28   0   0   0   1   0   0   0   1   0   0   1   0   0   0   0   1   0   0   0
## 29   2   0   0   0   0   1   0   2   0   1   0   2   0   1   2   2   0   0   0
## 30   2   2   0   0   1   0   0   0   2   2   2   0   0   0   0   1   0   0   0
## 31   2   0   1   0   0   2   0   0   0   0   0   1   0   0   1   1   0   2   2
## 32   2   2   0   1   0   2   2   1   0   0   1   0   2   0   2   2   0   0   0
## 33   0   0   2   0   0   2   0   0   2   1   0   0   2   0   1   2   1   2   0
## 34   0   2   1   2   2   2   0   0   0   2   2   1   0   2   0   1   0   0   2
## 35   1   0   2   0   0   0   2   2   2   2   1   0   1   0   1   0   0   0   1
## 36   2   0   0   1   1   0   0   0   1   0   0   2   0   2   0   1   0   2   0
## 37   1   2   1   2   2   2   1   0   0   2   0   2   0   0   1   2   1   1   1
## 38   0   2   2   1   1   2   2   0   2   0   1   0   0   1   0   0   1   2   1
## 39   1   2   0   2   0   0   0   1   1   0   2   2   0   0   0   2   2   0   1
## 40   0   2   0   1   0   0   2   1   0   0   1   0   1   2   2   0   0   0   0
## 41   2   0   2   0   0   2   2   0   2   0   1   0   0   0   0   2   0   0   2
## 42   0   0   1   1   0   0   0   2   1   0   1   0   0   1   0   0   0   0   2
## 43   0   1   0   1   0   1   0   1   1   0   1   2   0   0   1   1   0   1   0
## 44   0   2   0   2   0   0   0   1   0   0   0   2   1   0   2   0   2   1   0
## 45   1   1   0   2   0   2   1   1   1   0   2   1   0   1   1   1   1   0   2
## 46   0   0   2   0   2   1   1   1   1   2   0   0   0   0   2   0   2   1   0
## 47   2   0   0   0   2   0   0   0   0   2   0   0   0   2   0   0   0   1   2
## 48   0   0   1   0   0   0   2   0   2   0   1   0   1   0   1   2   0   0   0
## 49   0   0   1   0   1   0   2   1   0   2   0   1   2   0   0   0   2   0   2
## 50   1   2   0   0   0   0   0   1   0   0   0   0   2   0   1   1   0   2   0
##    X41 X42 X43 X44 X45 X46 X47 X48 X49 X50
## 1    2   1   2   1   2   0   2   2   0   0
## 2    2   2   1   0   2   1   0   0   0   1
## 3    1   0   2   2   0   0   0   1   0   1
## 4    0   0   0   2   1   1   0   0   2   0
## 5    2   2   1   2   2   2   1   0   2   2
## 6    0   2   0   0   1   2   0   0   0   0
## 7    0   1   0   0   0   2   1   0   1   0
## 8    0   1   1   1   1   0   1   1   0   1
## 9    2   0   2   2   0   0   1   0   0   0
## 10   1   2   0   0   1   2   1   0   0   1
## 11   2   0   2   2   1   2   1   1   0   0
## 12   2   0   0   0   0   2   0   2   0   2
## 13   1   1   1   0   1   1   0   1   1   0
## 14   0   0   0   1   0   0   0   0   0   0
## 15   0   1   0   2   0   0   0   0   2   2
## 16   2   0   0   0   2   1   1   0   0   0
## 17   2   0   1   1   0   1   0   1   1   1
## 18   0   1   0   0   2   0   2   2   1   2
## 19   0   1   1   0   0   1   2   1   1   1
## 20   0   1   0   0   0   2   1   0   0   0
## 21   1   1   0   0   0   1   2   1   0   0
## 22   2   0   0   0   0   2   2   0   0   2
## 23   2   0   0   1   2   0   0   2   0   1
## 24   0   0   0   2   0   1   0   0   0   0
## 25   0   0   0   0   0   0   0   0   0   2
## 26   1   1   0   0   1   2   2   0   0   0
## 27   0   1   1   0   0   0   0   0   2   0
## 28   1   0   2   1   0   2   0   0   0   2
## 29   0   1   1   2   0   0   2   2   0   1
## 30   1   0   1   1   2   1   2   1   0   0
## 31   2   0   2   2   1   0   0   0   0   0
## 32   0   1   0   1   0   0   2   1   1   1
## 33   0   0   0   0   2   0   1   0   0   1
## 34   0   0   2   2   1   0   0   0   1   2
## 35   2   2   0   0   0   2   0   0   0   1
## 36   2   1   2   2   0   2   2   0   0   0
## 37   0   0   1   0   1   1   0   2   0   0
## 38   1   0   0   1   0   1   0   0   1   0
## 39   0   2   0   2   1   2   0   2   0   1
## 40   2   1   0   0   0   1   1   0   0   1
## 41   0   0   0   0   2   2   2   2   1   1
## 42   1   2   2   1   2   2   0   0   1   0
## 43   1   1   1   2   1   0   0   0   1   0
## 44   0   0   0   2   0   2   0   0   0   2
## 45   0   0   1   0   2   1   2   1   1   0
## 46   2   0   0   0   0   2   0   2   0   2
## 47   1   0   0   2   0   0   2   2   0   2
## 48   2   1   2   2   0   2   1   0   1   0
## 49   0   0   0   1   1   0   2   2   1   0
## 50   2   2   0   1   2   0   1   1   2   1
```

- Randomly punch holes (i.e. NA) values in this matrix so that an each entry is missing with probability 30%.

```
n <- 50
R <- matrix(nrow=n, ncol=n, sample(c(rep(0, n*n*0.5), rep(1, n*n*0.25), rep(2, n*n*0.25))))

holes = matrix(nrow=n, ncol=n, sample(c(rep(0, n*n*0.7), rep(3, n*n*0.3))))

for(i in 1:n){
  for(j in 1:n){
    if(holes[i,j] == 3){
      R[i, j] = NA
    }
  }
}
R
```

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

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

```
order(rowSums(R, na.rm=TRUE), decreasing=TRUE)
```

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

- 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! This should be one line.

```
row <- apply(R, MARGIN = 1, sd, na.rm=TRUE)
col <- apply(R, MARGIN = 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. This should be one line.

```
apply(R>0, MARGIN = 2, sum, na.rm=TRUE)
```

```
##  [1] 16 16 19 14 17 25 17 16 14 15 18 19 15 17 20 22 19 15 24 17 19 18 18 19 15
## [26] 18 21 20 12 14 17 10 19 17 17 22 21 19 16 13 18 16 18 17 13 23 17 15 20 16
```

- 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`.

```
split(R, col(R))
```

```
## $‘1‘
##  [1]  1 NA NA NA  0 NA  2  2 NA NA  1 NA  0  0  1  0  2 NA  0  1  0  0 NA  0  1
## [26] NA  2 NA  2 NA NA  0  1 NA NA  1  2  1  1  0 NA  0 NA NA  2 NA NA  0  0  0
##
## $‘2‘
##  [1]  1 NA  1  1  0 NA  0  0  0  0 NA NA  0 NA NA  0 NA  2  1  0  1 NA  2 NA NA
## [26]  1  2 NA NA  0  1 NA  2 NA  1  2  0 NA  0  0 NA  2  0 NA NA  1  0  0  1  0
##
## $‘3‘
##  [1]  0  0 NA  0  1  2  2  0  2 NA  2 NA  0  1  0  2  0  2  1  0 NA  0 NA NA  0
## [26]  2  0  1  0  2 NA  1  1  2  1  0  0  0  0 NA NA  0  1  2 NA NA NA NA  1 NA
##
## $‘4‘
##  [1]  0 NA  1 NA  0  2  0 NA NA  0  0 NA  1  2  2  0  0  2  0  1  0 NA  2 NA NA
## [26] NA  0  2 NA  1  2 NA NA  1  0 NA NA NA  0 NA NA NA  0  1 NA  0 NA  2 NA  0
##
## $‘5‘
##  [1]  0  0 NA  0  0  1 NA  1 NA  0  0  0  2  1  1  2  2 NA  2  1  0  2 NA  0 NA
## [26]  0  0 NA  1  1 NA  2 NA  0 NA NA  1  0  0  0  0  0 NA  0  0  1 NA  1 NA  2
```

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

```
##
## $`33`
##  [1]  1 NA  0  2  0 NA NA  1 NA NA NA  2 NA  0  2 NA NA  2  0  1 NA  0  1  1  0
## [26]  0  2 NA  0  0  0  2  2 NA  2  0  2  0  0 NA NA  1  2  0  0  0  2  1  2  0
##
## $`34`
##  [1]  2  0 NA  0  2 NA  0  0  0 NA  0  0 NA  1  0 NA  1  0  2  0 NA  0  0 NA  1
## [26]  0  1  0  0  1  1  0  1 NA  2  0  2  2  0  1 NA  0 NA  1 NA  0  2  1  0 NA
##
## $`35`
##  [1]  0  0  0  0  0 NA  1  2 NA  0 NA  1 NA  2  2  0  0  0  0  0  1  1  0 NA  0
## [26]  0  0  2  0  0  2  0  2 NA  0 NA  2  1  1  0  1  2 NA  1 NA NA  2 NA  0 NA
##
## $`36`
##  [1]  0  0 NA NA NA NA  2  1 NA  1 NA  0  0  2  1  2  1 NA  2  2 NA  1  0  2  0
## [26]  2  1  2  0  0 NA  0  2  0 NA  1  2  0  0  0  0  0  2  1  1  2  0  1  0 NA
##
## $`37`
##  [1] NA  2  2  0  0  0  1  0  1  2  2  2 NA  2  1  1  0  0 NA  2  0  0  0  0  2
## [26]  0 NA  0  1  0 NA NA  1  0  2 NA  0  1  0  0  0  1  1  1 NA  2  2 NA NA NA
##
## $`38`
##  [1]  0  0 NA NA NA  1 NA  0  0 NA  1  2  2  2  0  1 NA  0  1  1 NA  0 NA  0  0
## [26]  2  2  0 NA NA  2  0  0  1  0  1  1 NA NA  0  2  2  0  0  0 NA  2  2  1 NA
##
## $`39`
##  [1]  1  0 NA  2  2  1  0 NA  0 NA  0  0 NA NA  2 NA NA  2 NA  2  2  0  0 NA  2
## [26] NA  1  2  2  0  2  0  2 NA  0  1 NA  1  0  0 NA NA  0 NA NA  0  0 NA NA  0
##
## $`40`
##  [1]  0  2  0 NA NA NA  1  0  1 NA  0 NA  0 NA  1  2  0 NA  0  0  0  0  0  2  2
## [26] NA  0 NA NA  1 NA NA  2  0  0 NA  0  2  2  1  0  0 NA NA NA  1  0 NA  0  0
##
## $`41`
##  [1]  1 NA NA  2 NA  0 NA  1 NA  0  0  2 NA  2  2  1 NA  1 NA  2  0 NA  0  1  1
## [26] NA  0 NA  0  1  0  0  2 NA  2 NA  2 NA NA  0 NA  2  0  1  0 NA  0  1 NA  0
##
## $`42`
##  [1] NA NA  1  0  0 NA  0  0 NA  1  0  2  1  0 NA  0 NA  0 NA  0  0  0 NA  2  1
## [26]  2  2 NA NA  1  0  2 NA  0  1 NA NA  1  2  2  0 NA  1  0  0 NA  2  0  0  0
##
## $`43`
##  [1] NA NA NA NA  0 NA  1  0  0  2 NA NA  2  1 NA  0  0  2 NA  2  2  2  2  1  1
## [26]  0  0  0 NA  0  0  0  1  2 NA  2  0  0  1  0  0  1  0 NA  1 NA  0  0  1  0
##
## $`44`
##  [1]  0  2  2  2 NA  0 NA  0  2  2 NA NA NA NA NA  2  0  0  0  0 NA  0  1 NA  0
## [26]  0  1  0 NA NA  2 NA NA  0 NA NA  1  1 NA  0  2  0  1  2  2  0  2 NA  1 NA
##
## $`45`
##  [1]  0 NA  0  1 NA  2  2 NA NA  0 NA NA  0  0 NA NA  0  2 NA  0  0  2 NA NA NA
## [26]  0  2  0  0  0  0  2  0  0  1  1  0 NA  1  0 NA  1  0 NA  0  2  2 NA NA  0
##
## $`46`
##  [1]  0  1  1 NA  2  0  2  1 NA  0  2  0  0  2  1  0  0 NA  2  0 NA  1  1  1 NA
## [26]  0  0  1  1  0  0  0  2  1  1 NA  1  1  1 NA NA NA  1  0 NA  2 NA  2 NA NA
##
## $‘47‘
##  [1]  1 NA NA  2  2 NA  0  1  1 NA  1  0  2  2  2 NA  1  0  0  1  0  0  2  0 NA
## [26] NA  0 NA NA  0  0  0 NA NA  0 NA  0  2  2 NA  2 NA NA  0  0  1  2  0 NA NA
##
## $‘48‘
##  [1]  0 NA  0  0 NA  0  0  0 NA  0  1 NA  0  1 NA  2 NA  2 NA  0  1 NA  2  1  0
## [26]  0 NA NA NA NA  2  1  2 NA  2  2  0  1  0 NA NA  0 NA  0  2  0  0  1  0  0
##
## $‘49‘
##  [1]  0  2 NA  0  1  0  2 NA NA  0  0  0  0 NA NA  2  2 NA NA NA  1  1  0  0 NA
## [26]  0 NA  1  1  0  2 NA  2  1  0 NA  1  2 NA NA  1  1 NA  1  1 NA  1 NA  1  0
##
## $‘50‘
##  [1]  0  0  1 NA  2  0  2 NA  0 NA  1  0  0  2  0  1 NA  1 NA  1 NA NA  0  0 NA
## [26]  0 NA NA NA  1 NA NA  1  0 NA NA  0  1 NA  2  2 NA  0  0  1 NA  2  0  1  0
```

- 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.

```
lapply(split(R, col(R)), function(R){c(min = min(R, na.rm = T),
           max = max(R, na.rm = T), pct_missing = (sum(is.na(R)) / n), first_NA =
           min(which(is.na(R))))})
```

```
## $‘1‘
##         min         max pct_missing    first_NA
##         0.0         2.0         0.4         2.0
##
## $‘2‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.36        2.00
##
## $‘3‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.28        3.00
##
## $‘4‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.42        2.00
##
## $‘5‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.28        3.00
##
## $‘6‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.24        5.00
##
## $‘7‘
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.36        2.00 
## 
## $‘8‘
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.38        7.00 
## 
## $‘9‘
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.32        1.00 
## 
## $‘10‘
##         min         max pct_missing    first_NA 
##         0.0         2.0         0.2         9.0 
## 
## $‘11‘
##         min         max pct_missing    first_NA 
##         0.0         2.0         0.2        21.0 
## 
## $‘12‘
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.36        1.00 
## 
## $‘13‘
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.38        1.00 
## 
## $‘14‘
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.18        2.00 
## 
## $‘15‘
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.28        4.00 
## 
## $‘16‘
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.32        2.00 
## 
## $‘17‘
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.42        4.00 
## 
## $‘18‘
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.36        1.00 
## 
## $‘19‘
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.26        4.00 
## 
## $‘20‘
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.28        7.00
```

```
##
## $`21`
##         min         max pct_missing    first_NA
##         0.0         2.0         0.2         2.0
##
## $`22`
##         min         max pct_missing    first_NA
##         0.0         2.0         0.3         1.0
##
## $`23`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.28        5.00
##
## $`24`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.32        1.00
##
## $`25`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.32        6.00
##
## $`26`
##         min         max pct_missing    first_NA
##         0.0         2.0         0.2         1.0
##
## $`27`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.28        4.00
##
## $`28`
##         min         max pct_missing    first_NA
##         0.0         2.0         0.2         1.0
##
## $`29`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.24        5.00
##
## $`30`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.38        3.00
##
## $`31`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.28        3.00
##
## $`32`
##         min         max pct_missing    first_NA
##         0.0         2.0         0.3         2.0
##
## $`33`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.28        2.00
##
## $`34`
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.24        3.00 
## 
## $`35`
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.24        6.00 
## 
## $`36`
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.22        3.00 
## 
## $`37`
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.22        1.00 
## 
## $`38`
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.28        3.00 
## 
## $`39`
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.36        3.00 
## 
## $`40`
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.34        4.00 
## 
## $`41`
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.36        2.00 
## 
## $`42`
##         min         max pct_missing    first_NA 
##         0.0         2.0         0.3         1.0 
## 
## $`43`
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.26        1.00 
## 
## $`44`
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.36        5.00 
## 
## $`45`
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.34        2.00 
## 
## $`46`
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.26        4.00 
## 
## $`47`
##         min         max pct_missing    first_NA 
##        0.00        2.00        0.34        2.00
```

```
##
## $`48`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.32        2.00
##
## $`49`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.34        3.00
##
## $`50`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.36        4.00
```

- Set a seed and then create a vector v consisting of a sample of 1,000 iid normal realizations with mean -10 and variance 100.

sd = sqrt(var) var = sdˆ2

```
set.seed(5)
n <- 1000
v <- rnorm(n, mean=-10, sd = sqrt(100))
```

- Repeat this exercise by resetting the seed to ensure you obtain the same results.

```
set.seed(5)
n <- 1000
v <- rnorm(n, mean=-10, sd = sqrt(100))
v
```

```
##    [1] -18.408554807863   3.843593434786 -22.554918626277  -9.298572335727
##    [5]   7.114408727024 -16.029079814547 -14.721663851669 -16.353713125243
##    [9] -12.857736348662  -8.618917751961   2.276303438535 -18.017794546528
##   [13] -20.803926000274 -11.575343561069 -20.717600398779 -11.389861405498
##   [17] -15.973130947129 -31.839667600916  -7.591827440633 -12.593554067343
##   [21]  -0.994880546667  -0.581306061323   4.679619034197  -2.932389104421
##   [25]  -1.809910697378 -12.934818487025   4.185890724859   4.987738274065
##   [29] -16.570820944857 -18.527954400020  -6.840849616385   1.096941676589
##   [33]  12.154605716780   2.171036389573   4.792217866383  -0.484261675821
##   [37] -20.095326459626 -30.004727386380 -27.621858724521 -11.426081259551
##   [41]   5.500603694831 -18.024231817148 -10.745789198827   8.956679547225
##   [45] -14.565689409204  -4.377766373570 -18.870085115114 -14.602445761952
##   [49] -17.243284860675 -10.692111558341   4.632485629487  -8.122739025669
##   [53]   0.220228613308 -15.918348329510 -11.122006550361 -19.249530858657
##   [57]  -2.466952017045 -11.126090702030 -10.640909282198  -7.667247064542
##   [61] -21.365828031485  -1.451695768715 -15.783704189619  -5.036384609698
##   [65] -17.600579306118 -13.413862703994 -31.023291204778 -13.017022813689
##   [69] -22.723834421812 -12.796661098092 -12.040973208196 -12.256141855174
##   [73]  -6.529715479779  -9.676321574021  -5.864687103282 -11.553484766254
##   [77]  -0.265146075170  -8.789098572265  -8.108263085225 -15.628850698260
##   [81]  -5.015838349987 -27.423024933631  -0.244709027963 -10.240828727364
##   [85]  -3.243155246859 -17.103096050534  13.872326463774 -14.734320121965
##   [89] -10.757725566668 -15.218400564783  -0.739528654376 -20.624111716142
##  [93]  -4.429661337020  -0.992694150878  -0.100543163118  -6.163919124203
##  [97] -13.465838136987 -15.401892500044 -11.825555932668 -10.592996499938
## [101] -29.953869678238   1.353112811710  -3.242054342462  -7.915167368927
## [105] -10.578456420851  -1.061885858974 -12.288653807756 -29.656526496405
## [109] -17.535104458236   2.801516244401 -19.529049597422   6.223793930094
## [113]  16.001420201224  -8.603514948780 -23.507196731275  -2.010689820521
## [117] -25.549958404453  -5.362799432060  -9.475704354268 -12.020318007205
## [121]   1.708564220070  -1.151551444280 -23.178886038772 -26.432509356051
## [125]   0.592503872462  -7.099164186113 -14.000334988939   2.430957780647
## [129] -23.664105180054 -24.414133018132   3.485490550278 -29.785283396939
## [133] -22.409505840471 -11.040391278101  -2.670270413566  -5.443203767402
## [137]  -7.119204524570 -20.736909106763  -3.512574605363  -7.008377214495
## [141] -17.959949930363 -10.293533971599  11.802357011983  -0.425815313656
## [145] -13.050486348849 -14.184033388889  -9.000459509964 -12.298096181618
## [149] -24.152148761828 -13.925988623094  -0.539114500504  -2.482291278866
## [153] -15.173768484891  -1.916640216054 -16.145352233324   2.382589282696
## [157] -13.380951424705   1.963663630902 -14.433183786245  -8.138851027497
## [161] -36.213448126526  12.462546198522  -9.065683189242   6.272800933659
## [165] -15.109175491651 -16.593808376624 -10.401901602379 -11.186940017694
## [169] -10.196568645371 -14.856784855791 -24.401475242738  -8.562311222607
## [173] -22.345866542670 -27.525012099284 -10.354962870507  -6.679650913982
## [177]   5.722882563282 -20.694705723933  -0.837134763853 -15.949928962835
## [181]  11.816466752786 -16.837732862980  -2.499407861985  -0.256173661449
## [185] -22.644734756613 -12.774214235846 -11.893986947641 -13.840249460645
## [189]  -2.594119768234 -21.683383911212  -3.324613018651  -6.337630503510
## [193] -15.149429938906  -5.494317600630 -11.877203797864   3.390693748472
## [197]  -1.837808157221  -9.177982358526 -16.508627214281  -2.735909823588
## [201] -11.136781772556 -12.951008265406  -0.108315308642 -17.751318058915
## [205]  -7.241017333831  -5.892183521632  -3.888168361215  -0.634292801086
## [209] -13.675417033988  -2.596232414025   2.185330550575  -3.708655658655
## [213]  -4.722536870171 -14.722553045383  -1.762848438316 -14.277882453683
## [217] -11.426439273501   4.187830490975  -5.128660987114  -3.965585489255
## [221]  -7.891671211629 -10.332992058426  10.251969893051 -13.707867497044
## [225] -25.782344495676 -11.215719524032 -27.966768164354 -14.755915430807
## [229] -18.841023211790 -44.980589839012 -13.819833688977  -0.223118749936
## [233] -15.580409458197 -16.264551466054 -15.304512253743   8.976215868765
## [237]   3.955406776193 -17.460258735475 -13.055730759423   1.696781686167
## [241]  -6.956128225256 -11.174982500921 -10.600855311923   4.709389470407
## [245] -24.781476069657 -16.836129460137  -5.394594008709 -11.815019251327
## [249] -21.588163079869  -5.909810824120 -12.582070558953 -12.668994392453
## [253]  -8.358440436223 -13.934589473704 -28.437372461671 -25.422882667314
## [257] -15.862403595313 -18.521389093399  -2.216754447079 -10.303170743814
## [261] -24.556575819274  -9.062150769358  -0.176507777429 -15.967101618036
## [265]  -9.251951493515  11.974294308226  -2.049767877962 -15.389422121212
## [269] -26.012831782189 -17.313735661773 -13.557407909462 -19.854143181581
## [273] -17.311706433707   4.653240833851   8.586153177076  -9.965029726201
## [277] -23.437752448970  -8.487059665727  -7.099908748756 -11.224783276633
## [281]  -8.748456118685 -17.724341139642 -20.129661227175  -0.330803991241
## [285] -14.233273003100 -18.315994547729   3.999572669764  -9.829860405126
## [289]   8.474969533898 -16.863393849524 -12.187240343770  -3.174070937309
## [293]  -4.758699754606  -9.194500215507  -9.462144575981 -17.372589881192
## [297]  -0.335619386235  -0.156931486725  -8.127090591315  -7.270142466753
## [301]   2.101460938081  -8.113428364998   9.624986717112  -8.612880607715
## [305] -25.786273543722 -17.970212711606   2.243538823335 -13.653335600032
## [309] -11.625902802538  -4.395208206223 -18.607256292097   2.386344918193
## [313]  -2.325642508412 -20.874091058541  -9.324962273272   6.051405650204
## [317]   2.322294300139 -13.791137947111 -23.498665837394  -6.350819003626
## [321] -13.635949245998   3.746533246060  -7.081042627373  -2.894082897278
## [325] -19.377609195422 -21.140631274282  -3.656280447981 -12.311929020707
## [329] -23.681942035459 -17.549074462007 -21.255966485424 -12.193592472707
## [333] -11.343079516958 -18.180206110392  -5.276658433326 -18.692561299721
## [337] -23.322883414694  -9.294371302955  -5.359068136384  -7.108415007645
## [341] -38.849410829094 -33.346917752388 -27.308910466768  -1.749904136181
## [345] -20.450395511171 -18.771933739183 -14.003898421516 -22.681889722295
## [349]  -8.614136094004   1.835716043282 -31.105550729875  -7.393238904631
## [353]  -0.544331743063 -16.199606197857 -10.091007394904  -4.797745772504
## [357]   8.062587078908 -29.125198979201  -8.007179249277  -7.235150469618
## [361] -18.362764273583  10.295974148845  -5.708940134388   0.639478660660
## [365] -16.058230530871   0.743730457831 -17.123476398959 -11.788601329005
## [369]  -5.002399609665 -11.995711405747  -9.101051954721   0.049100582958
## [373] -28.729416725919  -4.752154635982 -15.146273370969   2.109325695922
## [377] -15.127953770998   0.906256520853  -5.081551042632 -12.426289716873
## [381]  11.160472476042   1.886395727893  -1.047388093109  -1.314593629308
## [385]   6.561779651733   4.456359922537  -2.553995728922 -16.901902180113
## [389] -17.914117612371 -12.620811906477 -14.079171078140  -7.986889443606
## [393] -17.319474829679  -2.708642831478  -6.733513955663 -37.078778874531
## [397] -15.921389104291  -4.954563028428 -25.226970811374 -13.588751976189
## [401] -14.701264469396  -5.306776113894 -24.753401294403 -19.159547143038
## [405]  -5.420249427183 -17.125024603128 -13.784383285681  10.068609330801
## [409] -15.501683536525 -29.807861661627   0.345981307155 -18.448397163079
## [413] -20.169491169750  -3.252441711043 -10.332267760097  12.454914081593
## [417] -14.541469424969  -8.051254165231  -9.685019269209 -27.893309891810
## [421] -10.704909153343   8.041417550316  -5.553628752711 -21.909401000818
## [425]  -7.517708971977 -11.189781601893  -3.130110198244 -24.077777917780
## [429]  -4.246779592158  -3.622984696402 -19.481236861265 -25.155319715981
## [433]  -7.334618115042  -7.084677396126   7.089224384453  -5.232981523904
## [437]   3.075518037927   8.060126857840   0.353859900621  -4.834817028085
## [441] -13.941035209335 -25.102402298111  -9.940101634900  -1.449930087939
## [445]  -2.701485474450  -6.512411273018   3.613674698710 -12.886621289487
## [449]   8.884036638165 -10.173757697067  -4.589170876756 -15.400411124466
## [453] -10.808493963617 -10.296339578320 -11.452504839541 -10.108625955321
## [457] -23.787187646137  -3.284643234174 -21.542054532106  -5.774022139756
## [461] -13.869897261469 -22.405259848862  -5.430378549917 -24.107933534977
## [465]   9.339075276119  -4.304393465940  -0.672202481846 -25.453473369424
## [469]  -3.983816136527 -19.507661557502   3.603659375568 -12.665235899918
## [473] -40.349457705201 -12.388199142597 -12.721867185642   1.903968032762
## [477]  -9.488039073964  -6.689519660497  -4.683809026465  -3.859685312047
## [481]  -9.232841104419 -12.608792564282 -19.218704062694  -8.823938405802
## [485] -31.384102186566 -13.194195953844 -17.406944129704 -19.257914272400
## [489]  -2.420662111009  -6.404397365376 -18.292356751149 -14.457521220096
## [493] -16.709120468454 -18.009169023364   6.109466994028  -9.814816657695
## [497] -20.833993411097   2.894497568155 -11.439127356760 -32.880088377612
## [501] -10.847168184642  -8.009401734633   2.298258303962  -9.954474694292
## [505] -22.946076367869 -13.971613797066   7.394819880705   0.230209548892
## [509] -15.275672523016  -1.540038205486  -1.784506023688 -23.357693867781
## [513]   0.431180440569 -21.229909196730  -3.090767069332  -6.313944280955
## [517]  -3.514866383321 -14.398303143993 -18.664208547757  -1.346993973635
## [521] -15.098152029424 -17.587763236291  24.018720316697 -14.504159899519
## [525]   7.291981467503 -17.184882908888  -4.663804040837 -10.701869317107
## [529] -25.505635118521  -7.265724765598   8.852349774653  -1.568010033199
## [533]  -6.654660801656  -9.796145337848 -20.072915382416  -5.873060429184
## [537] -17.670323387486  -0.750081220333  -6.908505524954   9.668453631162
## [541] -19.375378001780  -5.795260240591 -15.392022371085  -1.314061215243
## [545]   1.501575193629  -9.823040000481 -19.278443348991 -15.057319445379
## [549]   2.257331789965   2.103777290379 -26.982737007954  -9.053036298059
## [553]  -2.378606599776  13.672035885614 -10.437028197029  -9.329019550200
## [557]  -8.447030792860 -15.328728859535 -28.289735439669  -3.994453735202
## [561]  -9.146599166658 -13.965735613055 -26.174143136255 -10.229052535902
## [565]  -5.937189753921  -4.909541302577   5.866710769106   0.463117755235
## [569] -20.479383075758 -17.906005273274  -5.327018087132  -4.461731387697
## [573] -16.809784054659 -37.949596029788 -20.845861259345 -16.570977988622
## [577] -19.692907479342  -5.872367980275 -19.154102857534 -14.483159992095
## [581] -17.985020115554 -20.172878560590   0.149889224187 -16.515076853809
## [585]  -9.617841505043  -8.159213331418  -5.482774136558  -6.921199352298
## [589]  -7.151417669762  -3.354489605479 -13.309747880980   4.915981333650
## [593]  -8.279097901407  -8.088397171910 -12.796150080915  -2.249804379654
## [597] -13.358452906988 -10.105294934623 -11.391350029848 -10.534659473403
## [601]   3.294961651692  -0.102306204240   6.645657872146 -31.198836484931
## [605]  -8.585191830716 -11.778062027229  -7.021160701696   1.394333480514
## [609]  -8.732624674773 -14.859502570436 -20.038316908649   7.842031984627
## [613] -18.671445586096 -19.705747848630 -22.028747828404 -13.673134964817
## [617]  -9.862263851585   3.200887098622  -5.107263260285 -22.023802675740
## [621] -18.735571674674  -8.880459645784   5.142313799774  -6.234401625474
## [625]   2.972814416908  15.623565618611 -20.020246553487  -1.993326717485
## [629] -20.359881686346   3.658774887000 -12.292210542837 -12.678130939206
## [633]  -6.263089631846  -0.194644468252  12.812914197663  -4.676750338048
## [637]  -5.011692111439  -8.458994996661  -9.172267479091 -25.859772889978
## [641] -25.295554252722   7.241555082112  -7.954306983572   2.378090102766
## [645]  -7.570362678311   0.135436534164   2.824896504923  -7.107908373717
## [649] -34.036859548940  12.685936883252 -16.110400290050 -14.634798238200
## [653] -24.817114469574 -13.041409548662  -6.867348179951 -12.650185419954
## [657] -23.603384688514  -6.055948493689  -4.612842337295  11.372919652853
## [661]  -7.927017802651 -22.766392792352 -15.156694473896 -10.788956438936
## [665]  -3.616515260365  -1.870749719277 -18.931847810246 -14.015955760547
## [669]  -1.758720501497 -23.040107074420  -1.267338542956  -3.102947320004
## [673] -18.533944947608  -3.458335309875   4.000088064126   0.024322885671
## [677] -40.421103237137 -12.422741267158   8.518098938743 -26.767815973311
## [681] -16.280176392912  -7.466663787544 -10.251887858082 -11.866206920128
## [685] -12.333797335240  17.242070482879 -37.925930102362   6.514171340088
## [689]  16.095467149863 -16.032468797240  -5.895687967058  -4.481324074596
## [693] -31.221467182943 -11.138993973786 -12.736714669705  -7.916229463028
## [697]  13.648679663872   6.055654175159   6.538888415793  12.142716243912
## [701] -14.411199113908  -4.366588265008   1.800894369027 -19.337733255350
## [705] -10.220070940028 -12.332465623732 -22.241461178005  -4.356858566067
## [709]  -2.818783808814 -17.719933377236  -5.657666674902  -7.038387659751
## [713] -19.716477060758 -16.409021918934  -0.770779019794 -20.862925740727
## [717]  -5.399093544114 -12.181479580269   3.230110657514  -7.146972360587
## [721] -18.944713662264 -10.177601131827  -5.776970278089  -8.766292886752
## [725] -26.584772297735 -11.278652462568 -18.905803071974  -1.030145798301
## [729]   3.509293665721 -20.320529450626  -4.863453356274  -6.602905652241
## [733] -12.641078638768  -5.548974845195   3.680193322704 -18.215119775135
## [737]  -8.086813615118   4.468181155409 -20.034287706975  -3.846267831978
##  [741]  -7.554143380086 -17.053527344416 -25.612190303246 -17.577140980839
##  [745]   5.420985101659 -10.646698990415 -19.665788418222  -6.241329850320
##  [749] -30.208101704324  -7.552074038858  -9.756425653812 -19.374108395415
##  [753]  -9.901249334498 -22.189417559531  -3.093728894419  -5.389919279314
##  [757]  -4.341355090294 -10.423638812117  -5.055078841773 -17.731818920097
##  [761]   5.808632570371 -28.251635806615  -6.731248085611 -12.806893456551
##  [765]  -1.019028262100 -16.415268103202  -8.587992346143 -26.545119305701
##  [769] -29.847284995526 -15.143258630465 -29.632814120709  -6.968309184765
##  [773]  -3.862362760284  -2.513224422317 -10.257374865110  -2.830517352809
##  [777]   0.791183585726 -11.236028205475  -2.703615772258  -7.900544363708
##  [781]  -0.126865971715 -25.164391932070 -10.183597847696 -15.893782774950
##  [785]  -7.937441086865  -5.478574360962  -3.339473776547 -15.269770880587
##  [789]  -9.748715897331  -4.480045336666  -5.849951594067  -6.459733582585
##  [793] -17.645999604513 -15.090991700472 -27.974375427724 -12.999655915267
##  [797] -14.502200448503  -0.736443778746 -24.013414444284   2.660777704727
##  [801] -12.375659729331 -19.143035944637   4.433643263402 -10.137015828307
##  [805] -15.303973497967  -1.902002531225 -13.352266694509 -24.391453516803
##  [809]   8.612932681250  -9.468142390273  -9.516362810552 -22.963058923234
##  [813] -20.720001137308 -10.205322504318 -17.084907882769   0.239477921104
##  [817]   0.823540795377  -4.452318743184   5.867799611694  -7.411390543521
##  [821]   2.651924760247   1.552432200891   6.734298576686  -3.703427915653
##  [825] -21.744450796905  -2.010780246827   2.099617836449 -15.448536988467
##  [829]   1.828370377876 -15.561691196066  -2.899575258112  10.743792051553
##  [833] -10.826767368597 -21.914410199602 -36.047961091743  -5.821087811478
##  [837]  -8.680307204318  -5.019925036336  -7.829476394510  -9.462918613658
##  [841] -10.635781108375 -17.322682002783   5.000586002350 -13.028308593571
##  [845]  -1.463747951267 -14.017487565080 -17.135859132682 -31.314101565976
##  [849]  -4.547053749930  -7.825585383664  -8.448059062468   2.495368049536
##  [853]  -5.031143817014 -13.090767527393 -15.492283577197  -6.424481457692
##  [857]   0.833901863066  -1.588595251024 -12.644799240713 -18.447493446382
##  [861] -14.237137300021  -7.324035660125  -6.862123852009  -0.315767770831
##  [865] -15.208417915128 -10.057442968787  -0.471092510455  -2.842119975268
##  [869] -15.214794118812 -13.472529387124 -21.936159279325  -4.870341331769
##  [873]  -9.598631221377  -3.052657451894  -1.597482966325 -16.550403661603
##  [877]  -2.372743653998   1.974678740014 -31.780074326623 -11.491097394422
##  [881]  -7.549643593207 -12.109977115440  -6.934218098484  -7.832304721116
##  [885] -16.667213928941 -19.637463942729 -22.925099460087  -3.099440567736
##  [889]  11.973152177280 -10.992650993574 -15.730250621238 -18.768324080494
##  [893]   1.075259612299 -19.133941201756   2.256126445517 -21.532241350859
##  [897] -18.020892644438 -23.012467276107 -15.441864938798  -1.375043120950
##  [901]  -2.277471688496 -19.945219160078  -6.596628738195 -14.366415068312
##  [905]  -8.049476538734 -10.812179559896   4.042550445102  -3.385818240079
##  [909]  -4.700966514262   6.619893276999 -13.625181243840 -24.411371708217
##  [913]   6.385781997361 -29.233654056437  -6.902476781712  -1.712048612673
##  [917] -24.619868114573 -11.034585241324   1.319492290326 -16.055966719190
##  [921] -14.539842757298 -10.976120225501 -13.370737212744  -7.498059291761
##  [925]  -2.749034997568 -26.839391924638 -17.573406981580  -9.757758481836
##  [929] -29.311259936488  -6.685376591689 -13.760447098014  -2.640777495024
##  [933] -18.332662690701 -14.801246394800 -34.066179953337  -5.871069210814
##  [937] -21.743993170410  -8.592672372483 -19.623332012699 -15.041854334387
##  [941]  -4.987227322622   0.300179946046 -17.782200085523  -3.945983970232
##  [945] -21.992010695461 -25.587553720289  -6.746547644624  -4.571484309491
##  [949]  -6.955760090322 -16.837510735254  -8.992992276882  -1.457139455632
##  [953] -13.813669822219 -15.751944010333 -19.159624772940 -17.027351387257
##  [957]  -6.687128070614   5.253743754699 -17.291469450897  12.357632396401
##  [961]  -5.261372414875   6.140322759885  -7.748485990930 -17.722730105545
##  [965]  -7.962098379411   0.974219721905 -12.683949861151 -24.533882678573
##  [969]  -7.619204154395  -3.569402923712 -17.084300581899 -16.130426794882
##  [973]  -8.616704547317  -3.287248252000  -4.395097094312   3.009143447609
##  [977] -29.295576059374 -32.806828746086 -22.366465165706  -7.955842747390
##  [981]  -1.078687720968  -4.533541246811 -31.254993790641  -7.337477592592
##  [985] -11.165343963540 -15.442772126946 -13.017272689548 -10.421837281536
##  [989]  -3.140296893955 -21.069152693408  11.744637367121 -13.122772252072
##  [993]  -6.066473565624 -13.943691879243   8.395464570109 -30.294484206949
##  [997] -25.579031939248   5.155938158397 -20.100748519180   0.897216039149
```

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

```
avg_v <- mean(v)
avg_v
```

```
## [1] -9.82600541
```

```
se_v <- sd(v)/n
se_v
```

```
## [1] 0.010120151595
```

- Find the 5%ile of `v` and use the **`qnorm`** function to compute what it theoretically should be. Is the estimate about what is expected by theory?

```
fifth_percentile <- quantile(v, probs = 0.05)
fifth_percentile
```

```
##            5%
## -26.593924482
```

```
qnorm(0.05, mean = -10, sd = sqrt(100))
```

```
## [1] -26.44853627
```

- What is the percentile of `v` that corresponds to the value 0? What should it be theoretically? Is the estimate about what is expected by theory?

```
ecdf(v)(0)
```

```
## [1] 0.84
```

```
pnorm(0, mean = -10, sd = sqrt(100))
```

```
## [1] 0.84134474607
```