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 $1 + 1/2 + 1/4 + 1/8 + \ldots$

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

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

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

```
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
## [26]  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] 0.54
```

- 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] 0.874
```

- 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 0 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
##   [38] 1 1 0 1 0 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
##   [75] 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 0 1 1
##  [112] 1 1 1 1 0 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1
##  [149] 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 0 0 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 0 1 1 1 1 1 1 1 1 0 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0
##  [260] 1 1 1 1 1 1 1 1 0 0 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
##  [297] 0 0 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 1 1 1 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 0 1 1 1 0 1 1 0 0 1 0 1 1 1 1 1 1 1
##  [371] 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1
##  [408] 1 1 1 1 1 1 0 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1
##  [445] 1 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 0 1 1 1 1 1 1 0 1 1 0
##  [482] 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
##  [519] 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [556] 1 1 0 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 0 1 1 1 1 1
##  [593] 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 1 1 0 1 0 1 1 1 1 0 1
##  [630] 0 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
##  [667] 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1
##  [704] 1 0 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
##  [741] 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 1
##  [778] 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 0 1 1 1 1 1 1 1 0 1 1 1 0
##  [815] 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 0 1 1 0 1 1 1 0 1 1 1 1 1
##  [852] 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 0 1 1 1 1 1
##  [889] 1 1 1 0 1 1 1 1 0 1 0 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 0 1 1 1 1
##  [926] 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 0 0 1 1 0 1 1 1 1 1 1 1 1 0
##  [963] 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1 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  felony      felony      none        felony      misdimeanor
##   [7] misdimeanor none        felony      felony      felony      infraction
##  [13] infraction  misdimeanor felony      none        none        misdimeanor
##  [19] infraction  none        misdimeanor felony      felony      misdimeanor
##  [25] felony      none        misdimeanor none        felony      felony
##  [31] infraction  infraction  misdimeanor infraction  misdimeanor none
```

```
##  [37] misdimeanor none        felony      none        infraction  felony
##  [43] felony      misdimeanor felony      misdimeanor felony      none
##  [49] infraction  infraction  misdimeanor none        felony      misdimeanor
##  [55] infraction  infraction  misdimeanor misdimeanor none        none
##  [61] misdimeanor felony      misdimeanor infraction  misdimeanor felony
##  [67] infraction  felony      none        felony      none        none
##  [73] felony      felony      infraction  none        none        infraction
##  [79] none        felony      felony      felony      none        none
##  [85] infraction  none        misdimeanor felony      infraction  none
##  [91] infraction  misdimeanor infraction  infraction  infraction  infraction
##  [97] misdimeanor none        felony      infraction
## 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  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
##  [13]  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
##  [25]  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
##  [37]  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
##  [49]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
##  [61]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE
##  [73]  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE
##  [85]  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [97]  TRUE FALSE  TRUE  TRUE
```

- 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  felony      felony      none        felony      misdimeanor
##   [7] misdimeanor none        felony      felony      felony      infraction
##  [13] infraction  misdimeanor felony      none        none        misdimeanor
##  [19] infraction  none        misdimeanor felony      felony      misdimeanor
##  [25] felony      none        misdimeanor none        felony      felony
##  [31] infraction  infraction  misdimeanor infraction  misdimeanor none
##  [37] misdimeanor none        felony      none        infraction  felony
##  [43] felony      misdimeanor felony      misdimeanor felony      none
##  [49] infraction  infraction  misdimeanor none        felony      misdimeanor
##  [55] infraction  infraction  misdimeanor misdimeanor none        none
##  [61] misdimeanor felony      misdimeanor infraction  misdimeanor felony
##  [67] infraction  felony      none        felony      none        none
##  [73] felony      felony      infraction  none        none        infraction
##  [79] none        felony      felony      felony      none        none
##  [85] infraction  none        misdimeanor felony      infraction  none
##  [91] infraction  misdimeanor infraction  infraction  infraction  infraction
##  [97] misdimeanor none        felony      infraction
## 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,]          0      1              0
##   [3,]          0      1              0
##   [4,]          0      0              0
##   [5,]          0      1              0
##   [6,]          0      0              1
##   [7,]          0      0              1
##   [8,]          0      0              0
##   [9,]          0      1              0
##  [10,]          0      1              0
##  [11,]          0      1              0
##  [12,]          1      0              0
##  [13,]          1      0              0
##  [14,]          0      0              1
##  [15,]          0      1              0
##  [16,]          0      0              0
##  [17,]          0      0              0
##  [18,]          0      0              1
##  [19,]          1      0              0
##  [20,]          0      0              0
##  [21,]          0      0              1
##  [22,]          0      1              0
##  [23,]          0      1              0
##  [24,]          0      0              1
##  [25,]          0      1              0
##  [26,]          0      0              0
##  [27,]          0      0              1
##  [28,]          0      0              0
##  [29,]          0      1              0
##  [30,]          0      1              0
##  [31,]          1      0              0
##  [32,]          1      0              0
##  [33,]          0      0              1
##  [34,]          1      0              0
##  [35,]          0      0              1
##  [36,]          0      0              0
##  [37,]          0      0              1
##  [38,]          0      0              0
##  [39,]          0      1              0
##  [40,]          0      0              0
##  [41,]          1      0              0
##  [42,]          0      1              0
##  [43,]          0      1              0
```

```
##  [44,]          0      0              1
##  [45,]          0      1              0
##  [46,]          0      0              1
##  [47,]          0      1              0
##  [48,]          0      0              0
##  [49,]          1      0              0
##  [50,]          1      0              0
##  [51,]          0      0              1
##  [52,]          0      0              0
##  [53,]          0      1              0
##  [54,]          0      0              1
##  [55,]          1      0              0
##  [56,]          1      0              0
##  [57,]          0      0              1
##  [58,]          0      0              1
##  [59,]          0      0              0
##  [60,]          0      0              0
##  [61,]          0      0              1
##  [62,]          0      1              0
##  [63,]          0      0              1
##  [64,]          1      0              0
##  [65,]          0      0              1
##  [66,]          0      1              0
##  [67,]          1      0              0
##  [68,]          0      1              0
##  [69,]          0      0              0
##  [70,]          0      1              0
##  [71,]          0      0              0
##  [72,]          0      0              0
##  [73,]          0      1              0
##  [74,]          0      1              0
##  [75,]          1      0              0
##  [76,]          0      0              0
##  [77,]          0      0              0
##  [78,]          1      0              0
##  [79,]          0      0              0
##  [80,]          0      1              0
##  [81,]          0      1              0
##  [82,]          0      1              0
##  [83,]          0      0              0
##  [84,]          0      0              0
##  [85,]          1      0              0
##  [86,]          0      0              0
##  [87,]          0      0              1
##  [88,]          0      1              0
##  [89,]          1      0              0
##  [90,]          0      0              0
##  [91,]          1      0              0
##  [92,]          0      0              1
##  [93,]          1      0              0
##  [94,]          1      0              0
##  [95,]          1      0              0
##  [96,]          1      0              0
##  [97,]          0      0              1
```

```
##  [98,]          0      0              0
##  [99,]          0      1              0
## [100,]          1      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 1 0 1 1 1 0 1 1 1 1 1 1 1 0 0 1 1 0 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1
##  [38] 0 1 0 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 1 0 0 1 1
##  [75] 1 0 0 1 0 1 1 1 0 0 1 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1
```

- 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
##             24             29             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    16.5353313140 -5.889145680703    7 0.05451974619387    2    0
## Emma      18.6287718314  8.241709400900   11 0.05944279405392    3    0
## Olivia    12.8584576775  9.136617011391    3 0.06527022174042    3    0
## Ava       14.8106097158 -0.351791479625    7 0.04710255961658    5    1
## Mia       16.7642225024 -5.125777767971    7 0.04934766505741    1    1
## Isabella  11.2141659599 -3.500251909718    4 0.18722487716106    3    0
## Riley     26.5582553029 -4.853582796641    6 0.03979760346313    1    0
## Aria      21.3588899330 -0.910677430220    6 0.02125210423643    3    0
## Zoe       22.3819313347  4.003533283249    5 0.01130979993482    3    0
## Charlotte 11.8407285126  0.880852839909    7 0.07247291489815    6    0
## Lily      13.3780790116  1.447487585247   12 0.03001452728899    5    0
## Layla     17.7757297081 -8.615727331489    6 0.19281188809666    0    0
## Amelia    16.1455270075  1.940662283450    7 0.05958528325169    3    1
## Emily      6.4006172580  0.778969512321    9 0.11639426430841    1    0
## Madelyn    7.3141866224 -1.004038695246    1 0.02345800099687    3    0
## Aubrey    22.7809436029  1.108957962133    7 0.12583927799377    2    0
## Adalyn    12.2305901640  1.691098799929    2 0.08429498949346    3    0
## Madison   21.6486469852  2.731775683351   14 0.00533140792959    3    0
## Chloe     18.4121094050 -8.368902863003    6 0.13264168260359    1    0
## Harper    14.0997186848 -3.041957728565    8 0.36499364341087    1    0
## Abigail   17.5962184607 -9.713989719748   12 0.12623113496797    6    1
## Aaliyah    9.5126343064 -6.648808987811    5 0.07256828605508    2    1
## Avery     21.1202026710  0.135829173960    5 0.10911900056177    5    0
## Evelyn    10.2614819253  2.317361203022    6 0.00131621394295    3    1
## Kaylee    16.8882152058  1.036147391424    4 0.02540915369947    3    0
## Ella      19.4217038451  8.221597508527    7 0.06707211542461    4    0
## Ellie     18.9120793627 -1.736004846171    7 0.16617569841370    3    1
## Scarlett  11.4480728428  2.438013660721    8 0.11020218642583    1    1
## Arianna   16.8191965168  8.185551180504    7 0.14190597779996    2    0
## Hailey    11.8833358965  7.507100882940    3 0.14022371317311    4    1
## Nora      16.6650282616 -9.138274155557    3 0.08021174381124    1    0
## Addison   15.0230555410 -3.264199197292    4 0.03430048748851    3    0
## Brooklyn  21.6299371106  8.088806224987    9 0.00621242479732    4    0
## Hannah    21.0351132099  3.294619107619   10 0.03902128499208    2    0
## Mila      16.9502279235 -1.055333181284    8 0.07895446883484    2    0
## Leah      11.9541762184 -6.293462687172    3 0.00782933117201    3    1
## Elizabeth 18.3798021592  4.443425214849    4 0.06057220392456    1    0
```

```
## Sarah     14.1672514585 -3.425864921883    4 0.03123666693739    1    1
## Eliana    17.3968770918 -8.375458908267    6 0.02249111612845    2    1
## Mackenzie 16.7940626273  8.933386509307    4 0.05990574824520    2    0
## Peyton    15.3509353323  0.930276284926    3 0.04075815762951    1    0
## Maria     18.1632355861 -8.365481011569    8 0.00389838016902    3    0
## Grace     16.7683624055  2.447256888263    4 0.03576581499334    3    1
## Adeline   21.4824907919 -3.977507068776    8 0.01764678111714    0    0
## Elena      4.2516323221 -1.163477762602    7 0.05601399350497    2    0
## Anna      22.3593715196  6.850816560909    4 0.01880855860351    1    0
## Victoria  13.3923888279 -5.003220993094    4 0.97195921749898    2    0
## Camilla   17.9746213672  5.335097853094    6 0.05001867730862    2    0
## Lillian   17.9922382274  1.220383853652    5 0.00542206075897    4    1
## Natalie    8.4889575152  8.675483758561    5 0.05822257542362    2    1
## Jackson   11.1131301286 -7.962810439058    5 0.13249668933812    2    1
## Aiden     16.8718470157  0.176246962510    3 0.14803033331977    1    0
## Lucas     24.8781451127 -0.334498933516    1 0.21918016319368    4    0
## Liam      32.0502223383  3.887115861289    9 0.25943623212732    2    0
## Noah       8.3900110823 -7.265367386863    7 0.10150004875468    1    1
## Ethan     25.2200749705 -3.943460551091    5 0.03281901694006    2    0
## Mason     32.1871389699  3.765230635181    5 0.07165683081581    0    0
## Caden     29.7139332048  1.165003175847    6 0.19755529862412    2    0
## Oliver    23.7067198667 -1.198640493676    4 0.29317043906540    2    0
## Elijah    22.4907502995 -5.166496983729    4 0.00036628228716    1    0
## Grayson   18.9660524585 -2.169358753599    7 0.00319192165034    3    0
## Jacob     23.6933159854 -7.179703763686    4 0.08858767411343    2    0
## Michael   22.4160328959 -6.265504877083   10 0.38575567414926    2    1
## Benjamin  17.5275872191 -0.178782381117    3 0.24252591583525    3    0
## Carter    12.2053865048 -5.474619064480   10 0.00484013139511    3    0
## James     20.1875763731 -3.723075403832    6 0.02794454875877    3    0
## Jayden    18.1030992620 -6.143584414385    6 0.04991311766207    2    0
## Logan     13.3103214287 -9.456956535578    6 0.44624277181245    2    0
## Alexander 22.5310500918 -8.475240082480    6 0.00276003753146    2    0
## Caleb     13.2545779722 -6.553712971509    6 0.15129489213175    5    1
## Ryan      12.0979928976  8.101327391341    5 0.00521040578476    1    0
## Luke      13.5621080015  9.444659696892    6 0.01934711733419    1    0
## Daniel    16.9556762850 -8.540663323365    5 0.02113368926156    1    0
## Jack      19.0489841923  9.151261774823    8 0.03471094214668    0    0
## William   20.7689773467 -7.495714868419   11 0.01282923843208    1    0
## Owen       2.6937599409 -3.653596066870    7 0.08238880190945    2    0
## Gabriel   18.4023889845  3.997475588694    5 0.01489672220002    4    1
## Matthew   22.1139378846  5.716171776876    4 0.10887212667975    4    0
## Connor    10.5980918979 -0.067766280845    5 0.10852361180249    2    0
## Jayce      9.9428731269 -1.396915921941    6 0.08233820170525    4    0
## Isaac     17.4071809304  0.449102474377    7 0.39940416375946    2    1
## Sebastian 26.3075783484 -6.990779158659    7 0.09233788338334    4    1
## Henry      6.8261800921 -9.013343746774    7 0.00876010192424    3    0
## Muhammad  10.6676394377 -9.094107318670    7 0.17205736305998    0    0
## Cameron   20.3353662718  0.045007974841    2 0.07528235913358    5    0
## Wyatt     21.2502921728 -0.699573219754    8 0.01319046905733    1    0
## Dylan     21.0696339289 -1.833642991260    7 0.05435637472611    2    0
## Nathan    13.9071056609  4.740027412772    7 0.10976312050336    1    0
## Nicholas   5.2935783528  3.798804166727   11 0.25949560994331    6    0
## Julian    22.1447702046 -5.492609860376    6 0.16169783306513    4    0
## Eli       11.5408645292  7.540391222574    5 0.09340489427610    1    0
```

```
## Levi      16.9941327039  3.939312458970    6 0.15215353201608    2    0
## Isaiah    24.7523747417  0.380588541739    5 0.00830033048987    1    1
## Landon    25.9428246740 -6.806413950399    5 0.01182911393485    3    0
## David     22.2678963636 -5.892560938373    6 0.04677441902459    0    1
## Christian 22.4582535042 -4.690786586143    8 0.15235232944237    1    0
## Andrew    15.0827798166 -5.815240866505   10 0.25530566956689    4    0
## Brayden   24.1793192771  7.938748635352    5 0.04619985804884    2    0
## John      14.0875255955  3.911257870495    5 0.05677752408692    2    0
## Lincoln   12.8331174630  9.867169861682    7 0.10624197735502    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.   : 2.6937599   Min.   :-9.71398972   Min.   : 1.00
##  1st Qu.:13.1555479   1st Qu.:-5.57326761   1st Qu.: 5.00
##  Median :17.1955049   Median :-0.52568235   Median : 6.00
##  Mean   :17.1357060   Mean   :-0.68760744   Mean   : 6.14
##  3rd Qu.:21.3897901   3rd Qu.: 3.77362402   3rd Qu.: 7.00
##  Max.   :32.1871390   Max.   : 9.86716986   Max.   :14.00
##        X4                      X5             X6
##  Min.   :0.00036628229   Min.   :0.00   DOMESTIC:76
##  1st Qu.:0.02492136552   1st Qu.:1.00   FOREIGN :24
##  Median :0.06023897608   Median :2.00
##  Mean   :0.10007782229   Mean   :2.37
##  3rd Qu.:0.12779752356   3rd Qu.:3.00
##  Max.   :0.97195921750   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  0  1  2  0  2  0  0  0   2   0   1   1   2   0   0   1   2   0   2   2
## 2   0  0  1  0  2  0  2  1  0   0   0   1   1   0   0   1   0   1   2   1   1
## 3   1  2  2  0  2  2  2  0  0   2   2   0   1   1   2   0   0   1   2   2   1
## 4   1  1  0  0  0  0  1  2  0   0   0   2   0   1   0   1   1   1   0   0   0
## 5   1  2  0  0  2  2  1  1  1   0   0   2   0   2   2   0   0   2   1   0   0
## 6   0  1  1  1  0  0  1  2  0   2   0   0   1   2   1   0   0   0   0   2   2
## 7   0  0  0  0  2  1  2  1  2   2   0   0   0   1   1   0   2   0   2   1   2
## 8   2  0  0  0  2  1  0  0  1   0   0   2   0   1   0   2   1   2   1   2   0
## 9   2  1  2  1  0  0  2  2  1   2   2   0   1   0   0   0   0   1   0   2   1
## 10  0  0  0  0  0  0  0  0  1   1   0   2   0   2   0   1   0   1   0   0   2
## 11  1  2  0  0  1  2  2  0  1   0   0   1   0   2   1   1   1   1   1   0   2
## 12  1  0  0  0  0  2  0  0  1   1   0   2   1   0   1   0   2   2   0   1   0
## 13  0  2  1  2  0  2  0  2  0   2   2   0   0   0   0   0   0   2   0   0   0
## 14  1  0  0  0  0  0  0  0  0   2   0   2   0   0   2   0   1   2   2   0   0
## 15  2  0  2  2  1  1  0  0  0   0   0   0   1   0   0   2   0   0   0   0   0
## 16  1  1  1  0  2  1  2  0  0   2   0   0   1   2   1   0   2   1   1   0   1
## 17  1  2  2  1  1  1  1  2  2   1   1   0   0   1   0   2   1   2   0   0   0
## 18  1  0  2  1  0  0  0  1  0   0   1   0   0   0   0   2   1   0   2   0   2
## 19  2  0  0  0  0  0  0  1  1   1   1   0   0   0   2   0   2   2   2   0   0
## 20  1  2  1  0  1  2  0  2  1   1   1   2   0   0   1   0   1   1   0   0   2
## 21  2  0  0  1  2  0  2  0  0   2   1   0   2   0   1   0   0   0   2   2   1
## 22  0  1  2  0  0  0  2  1  2   0   2   0   0   1   2   0   0   1   1   0   2
## 23  0  2  1  0  0  2  0  0  2   0   0   0   1   0   0   0   0   0   0   0   0
## 24  1  0  0  1  2  2  0  0  0   0   2   2   2   0   0   1   1   0   0   0   0
## 25  0  2  0  0  1  0  0  0  2   0   2   0   0   1   2   2   0   2   2   1   2
## 26  0  1  1  1  1  2  0  0  0   1   0   1   2   0   0   1   1   1   1   1   1
## 27  0  0  1  1  0  2  0  1  2   0   1   0   1   2   0   0   1   2   1   1   2
## 28  1  1  1  0  0  0  0  0  0   2   0   2   0   1   0   0   0   0   0   1   0
## 29  0  1  0  0  0  1  1  0  0   0   0   0   0   1   0   2   2   0   0   0   0
## 30  2  2  0  2  0  2  0  0  0   0   2   2   1   0   1   2   1   1   0   0   2
## 31  2  1  0  0  0  0  0  0  0   2   0   2   0   0   1   2   0   0   0   0   0
## 32  2  0  0  1  1  0  0  2  0   2   0   1   0   0   0   1   2   0   0   0   0
## 33  2  0  0  2  0  1  0  2  0   1   0   0   0   0   0   2   0   0   1   0   0
## 34  2  0  1  0  1  0  1  2  2   0   0   2   0   1   0   0   0   1   0   2   2
## 35  0  0  0  2  2  0  0  2  0   0   0   1   0   0   0   2   2   2   1   2   0
## 36  0  2  0  0  2  0  0  0  1   2   0   1   1   0   1   1   0   2   0   1   1
## 37  0  1  1  2  0  0  0  0  0   1   0   0   2   1   1   2   0   2   2   0   0
## 38  1  1  0  0  2  0  0  0  0   0   1   0   0   0   2   2   1   0   0   1   1
## 39  0  0  2  0  0  0  1  0  0   2   0   0   1   0   2   2   2   0   1   2   0
## 40  0  2  0  2  0  2  0  2  1   0   0   0   2   2   1   0   0   0   1   2   0
## 41  0  0  0  1  2  0  2  0  2   0   2   2   2   0   1   0   0   0   0   0   1
## 42  0  0  0  1  2  0  2  1  0   0   0   0   1   0   0   0   2   0   1   2   2
## 43  2  1  1  0  0  0  0  0  1   2   2   1   0   0   0   2   0   1   2   1   0
## 44  1  2  0  1  0  2  1  0  2   0   0   0   0   0   1   0   0   2   0   0   1
## 45  0  2  2  2  1  2  0  0  0   1   0   2   1   0   0   0   2   0   0   2   2
```

```
## 46  0  2  0  0  0  2  0  2  2  0  1  0  1  2  1  0  0  1  0  0  0
## 47  0  2  2  0  2  2  1  2  2  0  0  1  2  2  0  2  0  1  1  2  0
## 48  2  0  1  2  1  0  2  2  2  2  1  0  0  0  0  0  1  0  2  1  0
## 49  2  1  1  0  2  0  1  0  0  2  1  0  0  1  0  1  1  1  2  0  1
## 50  0  0  0  1  2  2  0  1  0  2  1  0  1  0  2  1  2  2  1  0  0
##     X22 X23 X24 X25 X26 X27 X28 X29 X30 X31 X32 X33 X34 X35 X36 X37 X38 X39 X40
## 1     2   2   0   0   1   0   1   0   1   1   2   1   0   1   0   1   0   0   1
## 2     0   1   0   0   0   0   0   0   2   0   0   0   2   2   0   2   0   0   0
## 3     0   0   0   0   0   2   2   0   1   0   0   1   2   0   1   0   0   1   2
## 4     0   0   0   0   0   2   1   1   0   0   2   2   2   0   0   1   1   0   2
## 5     0   0   2   2   2   2   0   0   0   2   0   0   0   1   1   0   1   1   1
## 6     1   1   0   0   1   2   1   0   2   0   2   2   1   0   0   0   0   1   0
## 7     0   1   1   2   0   1   1   0   0   1   1   0   2   2   1   2   0   0   0
## 8     2   1   0   2   2   1   0   0   2   0   1   0   0   1   0   0   1   2   0
## 9     2   0   0   2   0   0   1   1   2   2   1   0   1   0   1   0   1   0   0
## 10    0   0   0   2   0   0   2   1   0   0   0   0   0   2   0   1   2   0   2
## 11    2   1   0   0   1   2   0   0   2   0   2   0   1   1   1   1   1   2   1
## 12    2   0   1   0   2   1   1   0   1   1   1   2   2   2   1   0   2   2   0
## 13    1   0   0   0   0   0   1   1   1   0   0   2   0   0   0   0   2   0   0
## 14    2   0   2   0   1   1   2   0   0   1   0   0   2   0   1   2   1   0   0
## 15    1   0   1   0   1   2   0   2   0   0   2   0   0   0   0   0   2   1   1
## 16    0   2   1   0   0   1   0   0   0   1   0   0   0   0   0   0   2   1   0
## 17    2   0   1   0   2   2   0   0   2   0   1   0   0   2   1   0   2   0   1
## 18    0   1   0   0   1   0   1   1   0   1   0   1   0   2   0   0   1   2   0
## 19    0   2   1   0   2   2   0   0   2   0   1   0   1   0   1   0   2   0   2
## 20    1   2   0   2   1   2   0   0   0   0   1   2   1   0   0   0   0   2   0
## 21    0   1   1   1   0   0   0   0   2   0   0   0   0   1   0   0   0   2   0
## 22    0   1   1   2   0   1   0   0   1   1   1   1   0   0   0   0   0   0   1
## 23    2   1   0   1   2   1   2   1   1   0   2   2   0   2   0   1   1   2   0
## 24    0   0   2   0   1   0   2   0   0   1   0   0   1   0   0   0   1   1   0
## 25    0   0   0   0   0   2   0   1   2   0   2   2   0   2   2   1   0   2   1
## 26    0   0   2   0   0   0   0   2   0   0   1   1   0   2   1   0   1   2   0
## 27    1   0   1   1   0   1   0   0   0   2   1   0   0   1   0   0   0   0   1
## 28    0   2   0   0   2   2   1   2   1   0   1   0   2   0   1   1   1   0   2
## 29    0   2   1   2   1   2   0   0   2   0   2   0   1   1   2   0   2   1   0
## 30    1   0   1   2   0   0   1   1   2   0   1   0   2   0   2   0   0   0   0
## 31    1   0   1   2   2   2   1   0   0   1   1   0   2   1   1   0   0   0   0
## 32    0   0   2   1   0   0   2   2   0   0   0   1   0   0   0   0   0   2   0
## 33    0   0   2   1   0   0   2   2   0   0   2   1   1   0   1   1   2   0   2
## 34    2   0   2   0   1   0   0   0   0   0   1   1   0   0   0   0   1   2   1
## 35    0   0   2   0   0   2   0   2   0   0   2   0   2   1   2   0   1   2   1
## 36    0   0   2   2   0   0   1   0   0   1   2   2   0   2   1   2   0   2   1
## 37    0   1   0   2   1   1   0   0   1   1   0   2   2   2   2   2   1   1   0
## 38    2   1   2   0   2   0   2   2   1   0   0   2   1   0   1   2   0   0   0
## 39    0   0   0   1   0   1   2   0   1   0   0   2   1   0   0   0   2   2   2
## 40    0   0   1   0   0   0   0   1   1   2   0   0   2   0   0   2   1   1   0
## 41    0   1   0   0   0   1   0   0   1   1   1   1   0   0   0   0   2   0   1
## 42    0   0   1   2   0   0   1   2   0   2   2   0   1   0   2   2   1   2   2
## 43    0   2   0   0   0   0   2   0   0   1   2   0   0   1   0   2   0   0   0
## 44    2   0   1   2   1   2   0   0   0   2   1   0   2   2   0   0   0   0   0
## 45    0   2   1   0   0   1   0   0   1   0   0   0   2   0   0   0   0   2   0
## 46    2   2   2   0   0   0   1   1   0   1   0   0   0   0   2   0   2   0   0
## 47    2   0   2   0   2   0   2   0   2   1   0   0   2   0   0   0   2   0   2
## 48    2   1   0   0   1   1   1   1   2   0   2   1   0   2   0   0   0   0   0
```

```
## 49   1   0   1   1   0   2   0   0   1   0   2   2   1   0   0   1   1   2   0
## 50   1   0   1   0   2   2   0   0   1   2   2   2   2   0   1   2   0   2   1
##    X41 X42 X43 X44 X45 X46 X47 X48 X49 X50
## 1    1   2   0   0   0   1   0   0   0   2
## 2    0   0   0   1   2   0   0   1   0   0
## 3    0   1   1   0   1   0   0   2   0   2
## 4    0   1   0   0   1   0   2   0   2   0
## 5    0   0   1   0   2   1   1   0   0   2
## 6    2   1   2   0   0   2   1   0   2   1
## 7    0   1   2   2   1   0   2   2   2   0
## 8    0   0   2   1   0   0   1   1   1   0
## 9    2   0   1   1   0   0   1   0   2   0
## 10   0   0   0   2   0   1   1   2   0   2
## 11   1   2   1   1   1   0   1   0   1   1
## 12   1   0   0   1   0   0   2   0   0   2
## 13   0   1   0   2   0   1   0   0   1   0
## 14   2   2   0   2   2   0   0   1   0   0
## 15   1   0   2   0   0   0   1   0   2   1
## 16   1   2   1   0   1   1   2   0   0   0
## 17   1   1   1   2   0   1   0   0   1   0
## 18   2   2   1   1   2   0   0   2   0   1
## 19   2   0   1   1   1   1   2   2   2   1
## 20   0   1   1   2   1   0   1   2   0   1
## 21   2   0   0   1   1   0   1   1   2   1
## 22   0   0   0   2   2   0   1   2   2   0
## 23   0   2   1   0   1   2   0   1   0   2
## 24   2   2   0   2   2   0   0   0   0   0
## 25   2   2   1   0   2   0   0   2   0   2
## 26   0   0   0   0   2   0   0   0   0   0
## 27   1   2   0   2   0   0   2   0   1   2
## 28   1   2   1   0   0   1   2   0   0   0
## 29   2   2   0   0   1   1   2   1   1   0
## 30   0   0   2   0   0   1   0   2   0   0
## 31   0   2   0   0   0   0   1   2   0   1
## 32   0   0   2   1   0   1   0   2   0   0
## 33   0   2   0   0   1   1   0   2   1   0
## 34   0   0   1   0   1   2   2   0   1   1
## 35   0   1   0   2   1   0   0   2   0   2
## 36   1   0   0   0   0   0   0   2   2   1
## 37   0   0   0   1   1   2   0   2   2   1
## 38   1   0   0   1   2   0   0   0   0   2
## 39   0   0   2   1   2   1   0   1   0   2
## 40   0   0   0   0   2   1   1   0   0   0
## 41   1   1   0   0   0   0   0   0   0   0
## 42   2   1   0   0   2   1   0   2   2   0
## 43   0   1   0   1   2   0   1   0   2   2
## 44   2   0   2   0   1   1   2   2   2   0
## 45   0   2   2   2   2   2   0   1   2   2
## 46   0   0   2   0   0   2   0   1   1   0
## 47   2   2   0   0   0   0   2   0   0   0
## 48   1   0   1   2   1   0   2   1   0   0
## 49   0   1   0   0   0   0   0   1   2   2
## 50   1   1   0   1   0   2   1   1   0   0
```

- 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,]   NA    0   NA   NA   NA    2   NA    2    0     0     0     0     1
##  [2,]    1    0    0    0    0   NA    2    2    1    NA    NA     1     0
##  [3,]    0   NA    1    0    1    0   NA   NA   NA    NA     1     0    NA
##  [4,]    1    0    0   NA    1   NA    1    1    1     2     0     1     2
##  [5,]   NA   NA   NA    2    0    0   NA   NA    0    NA     0    NA    NA
##  [6,]   NA   NA    0    1    1    0    0    0    0    NA     2     1     2
##  [7,]    0    2   NA    1    0    2    0    1    0     2     1     0    NA
##  [8,]    2    0    2    0    0    0   NA   NA    0    NA     1     2     2
##  [9,]    0   NA    1   NA    2    1    0    1   NA    NA     0     1     1
## [10,]   NA    1   NA    0    1   NA    2    0   NA     1     0    NA    NA
## [11,]    0    1    2   NA   NA   NA    0   NA   NA     2     0     0     0
## [12,]   NA    2    0    0   NA    2   NA    0    1    NA     0    NA     0
## [13,]   NA    0    0   NA    0   NA   NA    2   NA     2    NA     0     1
## [14,]    2   NA    0    0    2   NA   NA   NA    2    NA     2     0    NA
## [15,]    0    0    1    0    2    2    2    1    0     0    NA     2     0
## [16,]    0    2    0    0    0   NA    0   NA   NA     2     1    NA    NA
## [17,]    0   NA    1    0    1    2    1    0    0     1     1    NA     0
## [18,]    0   NA   NA   NA   NA    0    0   NA    0     1     0     1     0
## [19,]    1    0    0    2    2   NA    0    2    1    NA     0     0     1
## [20,]    1    0   NA    0    0    1    1    0    0     0     2     0     2
## [21,]   NA   NA    2    0    2    1    1   NA   NA     1     0     0     0
## [22,]    0   NA   NA   NA    0    0   NA    2    2    NA     0     2     0
## [23,]    2    0    0    1   NA    2    1    1   NA     0    NA     0    NA
## [24,]    2   NA    0    0   NA    2   NA    2    1     0    NA    NA    NA
## [25,]   NA    0   NA    1    1    0   NA    1    0     0    NA     0     2
## [26,]   NA   NA    0    0   NA    0   NA    1    2     0     0    NA     0
## [27,]    0    2   NA   NA    2    1    2    0    0     0     0     1     2
## [28,]    0    0    0    0    2   NA    1    0   NA     0    NA     0     1
## [29,]    0    0   NA    0   NA    0    1   NA    1     2    NA     2    NA
## [30,]    0   NA    1    0    2   NA    2   NA    2     0     0    NA     0
## [31,]   NA    0   NA    0    1    1    0    0    1     2     0     0     0
## [32,]   NA    0    1    0   NA    1   NA   NA    1     2    NA     2     0
## [33,]    1   NA   NA    0    2   NA    1   NA    2     1     0     0     1
## [34,]   NA   NA   NA    0    0    1    1    0    1     0     0     2     0
## [35,]   NA    0    2   NA    0    0    0    0   NA    NA    NA    NA     0
## [36,]   NA   NA   NA    0    0    0    2    0    2     2    NA     2    NA
```

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

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

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

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

```
## [49,]     0
## [50,]     1
```

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

- 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] 12 12 14 10 22 22 19 14 17 19 11 18 20 23 21 16 24 16 18 13 24 15 23 19 20
## [26] 19 16 20 16 16 17 19 18 26 14 16 16 23 20 17 18 25 18 16 16 14 16 23 21 17
```

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

```
##
## $`6`
##  [1]  2 NA  0 NA  0  0  2  0  1 NA NA  2 NA NA  2 NA  2  0 NA  1  1  0  2  2  0
## [26]  0  1 NA  0 NA  1  1 NA  1  0  0 NA  2  2 NA  0 NA  1  2 NA NA  2  1  2  2
##
## $`7`
##  [1] NA  2 NA  1 NA  0  0 NA  0  2  0 NA NA NA  2  0  1  0  0  1  1 NA  1 NA NA
## [26] NA  2  1  1  2  0 NA  1  1  0  2  0  0  2  2 NA NA  0 NA NA  1  1  0 NA NA
##
## $`8`
##  [1]  2  2 NA  1 NA  0  1 NA  1  0 NA  0  2 NA  1 NA  0 NA  2  0 NA  2  1  2  1
## [26]  1  0  0 NA NA  0 NA NA  0  0  0 NA NA NA  0  0  0  2 NA  0  0  0  0  0 NA
##
## $`9`
##  [1]  0  1 NA  1  0  0  0  0 NA NA NA  1 NA  2  0 NA  0  0  1  0 NA  2 NA  1  0
## [26]  2  0 NA  1  2  1  1  2  1 NA  2 NA NA NA NA  0 NA  1 NA  1  0 NA  0  0  0
##
## $`10`
##  [1]  0 NA NA  2 NA NA  2 NA NA  1  2 NA  2 NA  0  2  1  1 NA  0  1 NA  0  0  0
## [26]  0  0  0  2  0  2  2  1  0 NA  2 NA  0  0  1 NA  2 NA  1  0 NA  0  1  0  2
##
## $`11`
##  [1]  0 NA  1  0  0  2  1  1  0  0  0  0 NA  2 NA  1  1  0  0  2  0  0 NA NA NA
## [26]  0  0 NA NA  0  0 NA  0  0 NA NA  0  0 NA  1 NA NA  1 NA  0  0  0 NA  0  1
##
## $`12`
##  [1]  0  1  0  1 NA  1  0  2  1 NA  0 NA  0  0  2 NA NA  1  0  0  0  2  0 NA  0
## [26] NA  1  0  2 NA  0  2  0  2 NA  2  0  0  1  1  0  0  0  1  1  0 NA  2  0  0
##
## $`13`
##  [1]  1  0 NA  2 NA  2 NA  2  1 NA  0  0  1 NA  0 NA  0  0  1  2  0  0 NA NA  2
## [26]  0  2  1 NA  0  0  0  1  0  0 NA  2  1  2  0  0  0  2  0  2  0  2  1 NA  2
##
## $`14`
##  [1] NA  1 NA  2  0  0 NA  2  1  0  2 NA  1  1  2  0  1  2  0  1  1  0  0 NA  0
## [26]  0  2 NA NA  0 NA  2 NA  0  2  2  1  2  0 NA NA  2  0  1  0  0  1  2  0  1
##
## $`15`
##  [1] NA  1 NA NA  0  1  1 NA  0  2  1 NA  0  0  1  0  0  0  0  1  1 NA  0  2  1
## [26]  0  2  1  0  0 NA NA  2 NA  1  1  2 NA NA  0  2 NA NA  0  0  1  1  1  0  2
##
## $`16`
##  [1]  0 NA  1 NA  2  0  0  0  0 NA NA  0  2 NA  0  0  2 NA  0  2 NA  0  2  1 NA
## [26] NA  2  0 NA  0 NA  2 NA  0  0  0  0 NA  1  2  0 NA NA  0  1 NA  1  2  2  1
##
## $`17`
##  [1]  1  2  0  0  2  0  2  0  1 NA  2  0  0  0  0 NA  2  2  0  2 NA  2  0  2  0
## [26] NA  2  2  2  0  0  0 NA  2  1 NA  0  0  2  0  1  2  1  0  0  2  2  1  1  0
##
## $`18`
##  [1]  0  0 NA NA  2  2 NA NA  0  0  2  1  1  0 NA  1 NA  0  2  2 NA NA NA NA  0
## [26]  0  1  2  0  1  0  0  0  1 NA NA NA NA  1 NA NA NA  0  1 NA  2 NA  0  1  0
##
## $`19`
```

```
##  [1]  0  0  2  1  0  1  1  1 NA NA  0 NA NA NA  2  0  1  0  0  2  0 NA NA NA  1
## [26]  1  2  1  2  2 NA  0 NA NA NA NA  2  0 NA  0  0  2 NA NA  2  0  0  2 NA NA
##
## $`20`
##  [1]  0 NA  0  0 NA  2  0 NA  1  1  0  2  1  0 NA NA  1 NA  0  0  0  1  2  0 NA
## [26] NA  0  2  0  0  0  2 NA  1  0 NA  0 NA NA NA  1  0  0  0 NA  0  2 NA NA  0
##
## $`21`
##  [1]  1 NA  0  0  1  0  2  2  1 NA  1  2  1  2 NA NA  0  0 NA  0  0  0  2  1  0
## [26]  2  2  1  2 NA  1  1 NA NA NA  1 NA  0 NA  2 NA NA  0  2  1  2  0  2 NA  2
##
## $`22`
##  [1]  1  0  0  0  0 NA  2  0  1  0  0  1  2  2  0  0 NA NA NA  0  0 NA  0  0 NA
## [26]  0  1 NA NA  0  0 NA  0  0  0 NA  0  2 NA  1  1  1  0  1  0  1  1  1  0 NA
##
## $`23`
##  [1]  0  1  2  1  0 NA  1  1  0  1  1  1  0  1 NA  1  0  1 NA  0  0 NA  0  0 NA
## [26]  2 NA  0  0  0  2  0  1  1  1  0  0  1 NA NA  1  1  0  2  1  0  2 NA NA  2
##
## $`24`
##  [1] NA NA  0  2  2 NA  0  1  1  1  1  2  1  0  0  0 NA NA  0  2  2  0  0 NA  1
## [26]  2 NA NA  1 NA  0  0  1  1 NA  1 NA  0 NA NA  2 NA  0  0  2  0  2 NA  0  0
##
## $`25`
##  [1]  1  2  1  0  0  1  1  0 NA  0 NA NA NA NA  1  0  2 NA NA NA  0  0 NA  0  2
## [26]  1 NA  0 NA NA  1  1  0  1  0 NA  0  0  2  1  1  1 NA NA  2  2  1  0  1  0
##
## $`26`
##  [1]  2  1 NA  2  0  0  0  2  0  1  0 NA  0  0 NA  0  1  0 NA  1  1  0  1 NA  0
## [26]  2  0  1  2  0 NA  2  2 NA NA  1  0  0  0 NA NA NA  2  2 NA NA  0  2  1  0
##
## $`27`
##  [1]  1  1  2 NA NA  0  2 NA  1 NA  2  0 NA NA  2 NA  0 NA NA  1  0  0  1 NA NA
## [26] NA  0 NA  0  0 NA NA  2 NA  0  2 NA NA NA  1  1  0  2  2  0  0 NA NA NA  2
##
## $`28`
##  [1]  0  0 NA NA  0  2 NA  1 NA  0  0  0  2 NA NA NA  1  2  1  2  0  1  0  0 NA
## [26]  0  0 NA  1 NA NA NA  2 NA  0  0  1  1  2  0  1  0  2  2  1  0  2  2  0  1
##
## $`29`
##  [1]  0  0  0 NA  2  0  2 NA  1 NA NA  1  1  2 NA  0  0  0  0  2  0 NA  0  2  0
## [26]  0  0  0  0 NA NA  1  0 NA  2  2 NA  0  1  0  1  0  0  2  0 NA  1  1  0 NA
##
## $`30`
##  [1]  0  2  0  0  1  0  0  0 NA  1  0  1 NA  0  0  0 NA  1 NA  2  0  0  0  0 NA
## [26]  0  0 NA  1  2 NA NA  1 NA NA  0  2 NA  2  1 NA NA NA NA  2  1 NA  0  1  2
##
## $`31`
##  [1] NA  0  2  1  0 NA NA  0 NA  2  0 NA NA  0 NA NA  1  0 NA  1  0  0  1  0  1
## [26]  1  2  0  0  2  0  1 NA  0  2  2  0  1  0  0  2 NA NA  2 NA  0  2 NA  0  0
##
## $`32`
##  [1]  0 NA  2  1 NA  0  1 NA  1  2  2  1  0 NA  0 NA  0  2  0  2  1  0  2  0  2
## [26]  2  0  0 NA  1  0 NA  0  1 NA NA NA  0 NA  0 NA NA  0  0  1  2  1  1  0  0
```

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

```
##  [1]  1 NA  1 NA  1  0  0  1  0  0  2 NA  1 NA  0  0 NA  0  0 NA NA  0  0  2 NA
## [26] NA NA  1  0  2 NA NA  1  1 NA NA NA NA NA  0  0 NA  0 NA  2  2  0  0  0  2
##
## $`47`
##  [1]  0  1  2  0  0 NA  0  0  1 NA  0  0 NA  0  1 NA  0 NA  0  2 NA  1  0  2  1
## [26]  0  0  0  2  0  0  0  0  2 NA  0  2  0  1  1  0  2  1  2 NA NA NA  0 NA  0
##
## $`48`
##  [1]  0  0  0 NA NA  2  1  2 NA  1  1 NA  2  2  1  1  0  0 NA  2 NA NA  1  2  0
## [26]  0  0  0  0  0  1 NA  0  0 NA  0  1 NA  1  1  0  1  2  2  1  2  2 NA  0  1
##
## $`49`
##  [1]  0 NA  0  1  1  0  1  2  1  2  0  2  0  2 NA NA  0  2 NA  0  0  1  0  2  0
## [26] NA  2  2  2  2 NA  0  0  0  0  0  0 NA  2  2  0  1  0 NA  2  2 NA  0  1  0
##
## $`50`
##  [1]  1  0  0  0 NA  0  2 NA NA NA  2 NA  1  2  0 NA  2  0 NA  2  0  2  0 NA  1
## [26]  0  0  0  1  1  0 NA  0  1  0  0  0  2 NA  0 NA NA NA  1  0  0  2  1  0  1
```

- 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
##        0.00        2.00        0.44        1.00
##
## $`2`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.38        3.00
##
## $`3`
##         min         max pct_missing    first_NA
##         0.0         2.0         0.4         1.0
##
## $`4`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.32        1.00
##
## $`5`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.28        1.00
##
## $`6`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.32        2.00
##
## $`7`
```

```
##         min         max pct_missing    first_NA
##        0.00        2.00        0.36        1.00
##
## $`8`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.34        3.00
##
## $`9`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.34        3.00
##
## $`10`
##         min         max pct_missing    first_NA
##         0.0         2.0         0.3         2.0
##
## $`11`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.32        2.00
##
## $`12`
##         min         max pct_missing    first_NA
##         0.0         2.0         0.2         5.0
##
## $`13`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.22        3.00
##
## $`14`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.22        1.00
##
## $`15`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.26        1.00
##
## $`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        0.12       10.00
##
## $`18`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.38        3.00
##
## $`19`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.36        9.00
##
## $`20`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.32        2.00
```

```
##
## $`21`
##  min  max pct_missing first_NA
## 0.00 2.00        0.28     2.00
##
## $`22`
##  min  max pct_missing first_NA
## 0.00 2.00        0.24     6.00
##
## $`23`
##  min  max pct_missing first_NA
##  0.0  2.0         0.2      6.0
##
## $`24`
##  min  max pct_missing first_NA
##  0.0  2.0         0.3      1.0
##
## $`25`
##  min  max pct_missing first_NA
##  0.0  2.0         0.3      9.0
##
## $`26`
##  min  max pct_missing first_NA
## 0.00 2.00        0.26     3.00
##
## $`27`
##  min  max pct_missing first_NA
## 0.00 2.00        0.44     4.00
##
## $`28`
##  min  max pct_missing first_NA
## 0.00 2.00        0.26     3.00
##
## $`29`
##  min  max pct_missing first_NA
## 0.00 2.00        0.24     4.00
##
## $`30`
##  min  max pct_missing first_NA
## 0.00 2.00        0.32     9.00
##
## $`31`
##  min  max pct_missing first_NA
## 0.00 2.00        0.28     1.00
##
## $`32`
##  min  max pct_missing first_NA
## 0.00 2.00        0.26     2.00
##
## $`33`
##  min  max pct_missing first_NA
## 0.00 2.00        0.42     1.00
##
## $`34`
```

```
##         min         max pct_missing    first_NA
##        0.00        2.00        0.22        2.00
##
## $`35`
##         min         max pct_missing    first_NA
##         0.0         2.0         0.3         1.0
##
## $`36`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.26        3.00
##
## $`37`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.38        2.00
##
## $`38`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.16        7.00
##
## $`39`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.34        2.00
##
## $`40`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.32        1.00
##
## $`41`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.34        1.00
##
## $`42`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.28        2.00
##
## $`43`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.42        3.00
##
## $`44`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.36        1.00
##
## $`45`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.36        4.00
##
## $`46`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.38        2.00
##
## $`47`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.22        6.00
```

```
##
## $`48`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.22        4.00
##
## $`49`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.18        2.00
##
## $`50`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.26        5.00
```

- 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 = sd^2

```
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
```

- 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
```