Lab 1

Janine Lim

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)
pi
```

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

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

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

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

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

```
prod(1/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/seq(from =1, by =  1, length.out =387))
```

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

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

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

Is this answer *exactly* correct? No - this number will be extremely small and close to 0 but is not 0 because we experienced numerical underflow.

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

```
-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
## [18] -12 -14 -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
```

```
x
```

```
##  [1]           Inf 4.19180654858 4.17341725189 4.15464876786 4.13548512895
##  [6] 4.11590933734 4.09590327429 4.07544759936 4.05452163807 4.03310325630
## [11] 4.01116871959 3.98869253500 3.96564727304 3.94200336639 3.91772888179
## [16] 3.89278926071 3.86714702345 3.84076143031 3.81358809222 3.78557852143
## [21] 3.75667961083 3.72683302786 3.69597450568 3.66403300988 3.63092975357
## [26] 3.59657702662 3.56087679501 3.52371901429 3.48497958377 3.44451784579
## [31] 3.40217350273 3.35776278143 3.31107361282 3.26185950714 3.20983167673
## [36] 3.15464876786 3.09590327429 3.03310325630 2.96564727304 2.89278926071
## [41] 2.81358809222 2.72683302786 2.63092975357 2.52371901429 2.40217350273
## [46] 2.26185950714 2.09590327429 1.89278926071 1.63092975357 1.26185950714
## [51] 0.63092975357          -Inf           NaN           NaN           NaN
## [56]           NaN           NaN           NaN           NaN           NaN
## [61]           NaN           NaN
```

Comment on the appropriateness of the non-numeric values.

We get NaN values because we can't take the log_3 of negative numbers and so it becomes a NaN value. We also get Inf values because of the Inf values we have from the original data set. * Create a vector of booleans where the entry is true if `x[i]` is positive and finite.

```
!is.nan(x)
```

```
##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [12]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [23]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [34]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [45]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE
## [56] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
is.finite(x)
```

```
##  [1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [12]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [23]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [34]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [45]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE
## [56] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

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

- 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(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? The number of levels make sense because we are trying to cast 53 different numeric values as factor variables so they will each be their own level.

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

```
##  [1] Inf               4.19180654857877  4.1734172518943
##  [4] 4.15464876785729  4.13548512895119  4.11590933734319
##  [7] 4.09590327428938  4.07544759935851  4.05452163806914
## [10] 4.03310325630434  4.01116871959141  3.98869253500376
## [13] 3.96564727304425  3.94200336638929  3.91772888178973
## [16] 3.89278926071437  3.86714702345081  3.84076143030548
## [19] 3.81358809221559  3.78557852142874  3.75667961082847
## [22] 3.72683302786084  3.69597450568212  3.66403300987579
## [25] 3.63092975357146  3.59657702661571  3.56087679500731
## [28] 3.52371901428583  3.48497958377173  3.44451784578705
## [31] 3.40217350273288  3.3577627814323   3.31107361281783
## [34] 3.26185950714291  3.20983167673402  3.15464876785729
## [37] 3.09590327428938  3.03310325630434  2.96564727304425
## [40] 2.89278926071437  2.8135880922156   2.72683302786084
## [43] 2.63092975357146  2.52371901428583  2.40217350273288
## [46] 2.26185950714291  2.09590327428938  1.89278926071437
## [49] 1.63092975357146  1.26185950714291  0.630929753571457
## [52] -Inf              NaN               NaN
## [55] NaN               NaN               NaN
## [58] 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? Infinity or -Infinity becomes NA values when we cast x to integers.

```
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
## [24]  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  2  2  2  2  2  2  2  2
## [47]  2  1  1  1  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) & x>0]
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 xˆ2 from 0 to 1 with rectangle width size 1e-6.

```r
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.

```r
mean(sample(c(0,1), size=100, replace=TRUE))
```

```
## [1] 0.55
```

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

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

```
## [1] 0.918
```

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

```r
?rbinom
mean(rbinom(n=1000, size = 1, prob = 0.9))
```

```
## [1] 0.896
```

- 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.

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

```
## [1] misdemeanor infraction  infraction  infraction  infraction  felony
## Levels: felony infraction misdemeanor none
```

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

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

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

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

```
##   [1] misdemeanor infraction  infraction  infraction  infraction 
##   [6] felony      felony      none        infraction  misdemeanor
##  [11] felony      infraction  felony      misdemeanor none       
##  [16] felony      none        none        infraction  felony     
##  [21] felony      misdemeanor infraction  misdemeanor felony     
##  [26] misdemeanor misdemeanor misdemeanor none        felony     
##  [31] infraction  infraction  felony      felony      none       
##  [36] none        misdemeanor misdemeanor infraction  felony     
##  [41] none        felony      infraction  misdemeanor misdemeanor
##  [46] felony      none        felony      felony      felony     
##  [51] felony      felony      misdemeanor felony      felony     
##  [56] misdemeanor infraction  felony      felony      felony     
##  [61] felony      infraction  infraction  none        misdemeanor
##  [66] infraction  infraction  infraction  none        felony     
##  [71] none        felony      misdemeanor infraction  none       
##  [76] misdemeanor felony      none        none        misdemeanor
##  [81] felony      none        felony      infraction  felony     
##  [86] felony      infraction  none        misdemeanor infraction 
##  [91] misdemeanor misdemeanor felony      none        infraction 
##  [96] misdemeanor misdemeanor infraction  none        infraction 
## Levels: none < infraction < misdemeanor < felony
```

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

```
p=3
n=100
matrix_1 = matrix(nrow = n, ncol = p)
matrix_1[,1]=as.numeric(x_3=="infraction")
matrix_1[,2]=as.numeric(x_3=="misdemeanor")
matrix_1[,3]=as.numeric(x_3=="felony")
colnames(matrix_1) = c("infraction", "misdemeanor", "felony")
matrix_1
```

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

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

The sum of each row should be 0 or 1 because each value in the matrix should only only be a 0 or 1 value, and each row should only have at most one 1 value for either infraction, misdemeanor, or felony. If they're all 0's, then the sum would be 0 and that would mean the person has committed no crimes.

Verify that.

```
rowSums(matrix_1)
```

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

- How should the column sum look (in English)? Each column sum should be the sum of people who have committed an infraction, then the sum of people who have committed a misdemeanor, and the third sum should be the number of people who have committed a felony. The column sums altogether should add up to the number of people who have committed an infraction, misdemeanor, and felony.

#TO-DO

Verify that.

```
colSums(matrix_1)
```

```
##  infraction misdemeanor      felony 
##          25          23          33
```

- 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
p = 6
p1 = rnorm(n, mean = 17, sd = sqrt(38))
p2 = runif(n, min = -10, max = 10)
p3 = rpois(n, lambda = 6)
p4 = rexp(n, rate = 9)
p5 = rbinom(n, size = 20, prob = 0.12)
p6 = sample(rep(c(0,1), times = c(0.76*100, 0.24*100), size = 100))
matrix_2 = matrix(c(p1, p2, p3, p4, p5, p6), n,p )
rownames(matrix_2)=fake_first_names
matrix_2
```

```
##                     [,1]             [,2] [,3]            [,4] [,5] [,6]
## Sophia    21.3169727761 -0.096975346096    6 0.1068797099579    2    0
## Emma      19.6198830027  4.636536929756    5 0.0741292919136    3    1
## Olivia    11.4343812761  2.113138409331    5 0.1063200732026    1    0
## Ava       13.1021756953 -5.972205088474    4 0.0020688884817    5    0
## Mia       20.9698482808  8.777321930975    6 0.0496336620094    6    0
## Isabella   9.8480239700 -2.403197004460    0 0.0668853885598    3    1
## Riley     12.0741826443  8.093195031397    7 0.0642986858471    3    0
## Aria      19.3809076164 -7.162421280518   10 0.0840790860299    1    0
## Zoe        9.1963560853 -4.968998203985    8 0.1344232374963    1    0
## Charlotte 24.2542887551  6.605690754950    7 0.0181559816831    4    1
## Lily      12.6642489987 -2.173726879992    3 0.0597835512728    0    1
## Layla     21.9711575140  5.695860465057    8 0.1058250007132    5    0
## Amelia    20.0552294910 -0.374140175991    9 0.0510217141774    0    0
## Emily      9.6600227750  6.487084217370    6 0.0897818012116    3    0
## Madelyn    3.0008517254 -5.008912477642    9 0.1421927944193    2    0
## Aubrey    21.7108066494 -6.453156713396    9 0.0230014395072    3    0
## Adalyn    18.6232831435  7.391738751903    6 0.0557242075188    3    1
## Madison   22.6253124456  6.307426267304    6 0.1946437120758    1    0
## Chloe      9.1896896669 -7.077128943056   13 0.1559690630177    2    0
## Harper    14.5667366457 -5.746584222652   11 0.0830368822920    4    1
## Abigail   25.2478406522 -1.097740624100   10 0.0553441676829    5    0
## Aaliyah   21.9932940940 -3.232096279971    1 0.0629020892601    0    0
## Avery     12.1001894112  5.891855424270    6 0.0409601369562    2    0
## Evelyn     5.8618105567  0.567268277518    5 0.0648080106411    1    1
## Kaylee    15.8999223073 -2.418528804556    3 0.0499507752765    1    0
## Ella      22.4294133675 -3.776327325031    8 0.1000748329968    2    0
## Ellie     21.7752955201  9.434805074707    9 0.0018543044312    2    1
## Scarlett   5.2005529174  1.837662965991    5 0.0668993279752    2    0
## Arianna   25.6853270281  7.104822369292    3 0.0221808305439    2    0
## Hailey    17.7612387500  6.188832842745   11 0.2230191181010    0    1
## Nora      21.2480955629 -2.247545211576    9 0.2496409703734    5    0
## Addison   20.9457173028 -5.913596018218    5 0.2275897082262    1    1
## Brooklyn  14.6920121551  3.842462515458    3 0.1312907578170    1    0
## Hannah    18.0931991882  7.779223225079    6 0.0501357404929    2    0
## Mila      25.6525701116 -3.447828916833    5 0.0745909092948    1    0
## Leah      28.3350072010  8.807859770022    6 0.0482411537733    3    1
## Elizabeth 18.2701073594  7.505804151297    5 0.1213401187579    4    0
## Sarah     15.2972454977  6.026624310762    2 0.0488914123012    3    0
## Eliana    16.0286752011 -0.080457162112    4 0.2455739356023    1    0
## Mackenzie 22.2788053146  2.029335345142    9 0.3341913269475    0    0
## Peyton    10.9834109863 -1.177033935674    6 0.2274919268064    1    0
## Maria     20.7519528273 -0.098207639530    5 0.0301265249339    3    0
## Grace     14.6748630711 -7.974054622464    7 0.2355312620334    2    0
## Adeline   30.5604613402  8.335438342765    3 0.1295297257556    0    0
## Elena     28.6921833274 -2.852512164973    6 0.0095686897015    0    1
## Anna       8.0666221326 -3.210826884024    5 0.1556745243890    2    0
## Victoria  19.0686174386 -5.464843013324    6 0.0656315433379    5    0
## Camilla   24.2618287090 -4.660149314441   10 0.0308884397770    0    0
## Lillian   18.3301898105 -5.143688586541    2 0.0658115846001    2    0
## Natalie   27.0082787691  9.865225846879   10 0.0209475377471    3    1
## Jackson   26.1657126455 -7.269071363844    4 0.0337962295717    1    0
## Aiden     17.1527375217 -7.118680868298    7 0.0313095301448    1    0
## Lucas     19.0324436566  6.190170319751    4 0.0365863330145    1    0
## Liam      20.4698411702  6.511795665137    2 0.0140938474279    2    0
## Noah      20.8543005519 -5.909203304909   11 0.0499005918908    2    0
## Ethan     24.7485919448 -4.411310036667    6 0.0352618707771    5    0
## Mason     14.5484875133 -0.694920876995    6 0.2426496859129    1    0
## Caden     22.0404089354  1.062555839308   11 0.0512335383230    1    0
## Oliver    26.8917653152 -7.123933881521   10 0.1245190988559    3    1
## Elijah    23.9539813629  8.581551155075    7 0.0617041711489    2    0
## Grayson   14.4697878275  0.627228887752    7 0.1779532418415    1    1
## Jacob     17.4500475029  7.996172895655   10 0.2254431394560    2    0
## Michael   10.4960139692 -0.887140790001    8 0.1431217168722    2    0
## Benjamin  21.7626198636 -2.217747964896    5 0.1201887754677    3    0
## Carter    17.2666232121  1.130856024101    6 0.0185544244014    2    0
## James     15.5565211965 -8.435157751665    7 0.1339013988312    3    0
## Jayden    30.2636740205  9.609402297065    8 0.0920863127410    3    0
## Logan     22.5064479176 -0.656472281553   10 0.0123666708047    4    1
## Alexander 20.1061886577  6.263528424315    5 0.0512501986490    2    0
## Caleb     22.3765053657  2.730376068503    5 0.0845444917425    1    1
## Ryan      19.6908940988  6.782528911717    4 0.0279928166109    0    0
## Luke       6.4923264266 -8.859407706186    5 0.0822349184719    1    0
## Daniel     7.8478859219  4.927779510617   12 0.0119340078802    1    1
## Jack      15.2170071341  1.240738867782    4 0.0592677213976    6    0
## William    1.6020565062 -9.623404447921    2 0.2273875358939    6    0
## Owen      28.2860719476  1.493492764421    7 0.2286465372797    2    0
## Gabriel   11.3819359452 -8.743146401830    6 0.2022763538184    1    0
## Matthew   13.9465103707 -3.232774757780    5 0.1997518982231    4    0
## Connor    13.3103153410 -8.933860198595    5 0.0571907323061    4    1
## Jayce     19.3237194694 -5.443692808039    7 0.6661271502201    0    0
## Isaac     19.4659059627 -7.388098626398    2 0.1095398988373    1    0
## Sebastian 20.1672088738 -1.625044718385    7 0.0852538415495    3    0
## Henry     18.9097693100  7.118848874234    6 0.0975014184451    5    0
## Muhammad  18.4318262725  0.551281501539   12 0.3012923629194    2    1
## Cameron   21.9160602969 -9.236788768321    8 0.1603627314432    4    0
## Wyatt     11.8696840199  2.797510963865    7 0.0010312741824    4    0
## Dylan      9.4168019538  3.513419907540    6 0.0242790063947    4    1
## Nathan    18.8956792917 -5.019447100349    4 0.0933227507972    4    1
## Nicholas  15.5632242475 -1.909407330677    3 0.0562213121706    4    0
## Julian    15.1909890355 -9.438405893743    8 0.2917465736849    4    0
## Eli       11.0013998895  0.831363312900    6 0.0276798165397    3    0
## Levi      15.9267721327 -2.942378316075    6 0.0572537430045    3    0
## Isaiah    24.3775285922 -0.051264031790    4 0.1807487938046    4    0
## Landon     9.8060973955  8.938338411972    6 0.0808855425520    2    0
## David     15.7471039166 -2.004809840582   10 0.0754230283201    3    1
## Christian 18.5970988937  6.026346068829    3 0.1907861232017    1    0
## Andrew     4.8151344971 -6.303680115379    4 0.0244682714240    1    0
## Brayden   12.4629593121 -7.597714364529    9 0.1814626277228    2    0
## John      22.7932166083 -0.708377817646    9 0.0014854661810    3    0
## Lincoln   18.0021816198 -2.585894037038   11 0.8179805999652    1    1
```

- 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.

```
X_4 = data.frame(matrix_2)
X_4$p6_cat = factor(p6, labels = c("DOMESTIC", "FOREIGN"))
##View(X_4) I couldn't knit the file unless I took this view function out- sorry!
```

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

```
table(X_4$p6_cat)
```

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

```
table(X_4$p6_cat)/100
```

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

Print out a summary of the whole dataframe.

```
summary(X_4)
```

```
##        X1                    X2                     X3
##  Min.   : 1.6020565   Min.   :-9.62340445   Min.   : 0.0
##  1st Qu.:13.2582804   1st Qu.:-5.01154613   1st Qu.: 5.0
##  Median :18.5144626   Median :-0.67569658   Median : 6.0
##  Mean   :17.6102715   Mean   :-0.10353619   Mean   : 6.4
##  3rd Qu.:21.9298346   3rd Qu.: 5.92547809   3rd Qu.: 8.0
##  Max.   :30.5604613   Max.   : 9.86522585   Max.   :13.0
##        X4                      X5             X6             p6_cat
##  Min.   :0.0010312742   Min.   :0.00   Min.   :0.00   DOMESTIC:76
##  1st Qu.:0.0487288477   1st Qu.:1.00   1st Qu.:0.00   FOREIGN :24
##  Median :0.0750069688   Median :2.00   Median :0.00
##  Mean   :0.1122053568   Mean   :2.33   Mean   :0.24
##  3rd Qu.:0.1462599187   3rd Qu.:3.00   3rd Qu.:0.00
##  Max.   :0.8179806000   Max.   :6.00   Max.   :1.00
```

- 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(sample(rep(c(0:2), times = c(0.50*2500, 0.25*2500, 0.25*2500)), size = n*n), nrow = n, ncol =
table(R)
```

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

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

```
holes = matrix(nrow = n, ncol = n, sample(c(rep(0, n*n*0.7), rep(3, n*n*0.3))))
for(i in 1:n){
  for(j in 1:n){
    if(holes[i,j]==3){
      R[i,j]=NA
    }
  }
}
R
```

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

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

- 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.

```
apply(R, MARGIN =1, sd, na.rm=TRUE)
```

```
##  [1] 0.85215816722 0.83205029434 0.82807867121 0.87066900492 0.86711818075
##  [6] 0.75996059566 0.61220087877 0.93455856719 0.84281592351 0.90558034297
## [11] 0.79979754523 0.85588532090 0.89459504822 0.90556993049 0.90243777591
## [16] 0.84418225411 0.89928422716 0.75833704583 0.79716520920 0.80622577483
## [21] 0.84077140277 0.82285973943 0.82381956682 0.77459666924 0.79046271831
## [26] 0.66219534414 0.75807647576 0.81649658093 0.95600222596 0.85901293693
## [31] 0.80950789391 0.74775650111 0.87078025831 0.82686886579 0.89410905932
## [36] 0.84002688129 0.83738850732 0.83300806660 0.83757892854 0.90354816528
## [41] 0.87423435890 0.83816526318 0.88687914726 0.84316331143 0.73906595605
## [46] 0.93094933625 0.61220087877 0.88687914726 0.79789512885 0.81517858720
```

```
apply(R, MARGIN = 2, sd, na.rm=TRUE)
```

```
##  [1] 0.78072439645 0.82836355919 0.82182530102 0.85512059455 0.74907350181
##  [6] 0.85301950530 0.90632696717 0.84492824744 0.76523564056 0.78363384079
## [11] 0.81066855082 0.71771928199 0.89201957643 0.87376286169 0.89796949077
## [16] 0.90792308282 0.80546911040 0.89794560320 0.85901293693 0.84861216259
## [21] 0.85588532090 0.77459666924 0.89155582824 0.82285973943 0.82227511432
## [26] 0.87926630988 0.80024034851 0.87679459896 0.90486632647 0.84440066184
## [31] 0.81867681600 0.93642615266 0.79176634141 0.85208592300 0.80455691406
## [36] 0.88963130018 0.74721705905 0.75037528148 0.88616323851 0.81096094706
## [41] 0.84497248158 0.84497248158 0.83198088451 0.78933142388 0.68144538746
## [46] 0.90155858477 0.89810654458 0.76341057035 0.84091786587 0.90551882884
```

- 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] 18 15 17 21 19 23 16 18 10 21 21 11 18 21 17 18 16 19 17 14 19 15 13
## [24] 22 14 19 22 19 14 20 18 14 18 17 17 16 17 22 18 21 16 16 14 23 13 14
## [47] 15 20 18 14
```

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

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

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

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

```
##
## $`14`
##  [1]  0  0  0  1  1 NA NA  0  0 NA  0 NA  0  2  2 NA  2  2 NA  2 NA NA  2
## [24]  1  2 NA  0  1  0  0  2  0  0  2  1  2  0  0  2 NA  1  0  0  1  0  0
## [47]  1  2  0  1
##
## $`15`
##  [1]  0  2  0 NA  2 NA  2 NA NA  0  0  0  2  0  0  2 NA  1 NA NA  2  0 NA
## [24] NA  1  1  0  0  2  0  1  0  1  0  0 NA  2 NA  0  1  2  0  0  0  0  2
## [47]  2 NA NA NA
##
## $`16`
##  [1]  2  2  0  0  0  1  0  0  2  2 NA  2  1  0  2 NA NA  0  1 NA  0  2 NA
## [24]  1 NA NA  0  0  2  0  1  0  0  0  2  1  0  0 NA NA  2  0 NA NA  0 NA
## [47] NA  2  0  2
##
## $`17`
##  [1]  1  1  0 NA  0  2  0  2  0  2  0 NA  2 NA  0  0  0 NA  1  0 NA NA  0
## [24]  0  0  1  0 NA  0  0  0  1  0 NA  0  0  0  0  1  2  2  0  0  2  1  2
## [47]  1  0  0 NA
##
## $`18`
##  [1]  0 NA  1 NA  0  2  0  0  1  0  0 NA NA  0 NA  0  2  0  1  2  2  0  2
## [24]  1  0  0  2  1  2  2 NA  2  0  1 NA  1  2  2  2  0  0  0 NA NA NA NA
## [47] NA NA  0  0
##
## $`19`
##  [1] NA NA NA  0  1  0  0  1  0  2 NA  2  0  0 NA NA  0  2  1  1 NA  2  0
## [24]  2  1 NA  0  2 NA NA  2 NA  0  0  0 NA NA  0  2 NA NA NA  2  0  0 NA
## [47]  1  1 NA  1
##
## $`20`
##  [1]  1  0  0 NA NA  0  1  2  0 NA  0  0  0  2 NA NA NA  0  0 NA  1  0 NA
## [24] NA  0  2  0  0  2  0  0  2  0  2 NA  1 NA NA NA  0  0  1 NA  2  1  2
## [47]  0 NA NA  0
##
## $`21`
##  [1]  1  0  2  0  2  0  0 NA  2  1 NA  1 NA  2  0 NA NA NA  0  0  0  1  0
## [24]  2  1  1  0  0  0  0 NA NA NA  2  1  2  0  0 NA  0  2  1  0 NA NA  2
## [47] NA  1  2 NA
##
## $`22`
##  [1] NA NA  1  0 NA  0 NA  2  1  0  0  0  2  0 NA NA NA  0  2  0  1  0 NA
## [24]  1  2 NA NA  2  0 NA NA NA  0 NA  0  1  0  1  0  0  0  0  1  2  0  0
## [47]  1  1  0 NA
##
## $`23`
##  [1] NA NA  0  0  0 NA  1  2 NA  2  2  0 NA NA NA NA NA  1  2 NA NA  1  0
## [24]  0 NA NA  1  2  0 NA NA  0  2 NA  0  0  2  0  0  2 NA  1 NA NA  0 NA
## [47]  0 NA NA NA
##
## $`24`
##  [1]  0  0  1  0  0  1  0  2  1 NA  0  1  2  0 NA  0  2  2 NA  1  2  1  1
## [24] NA NA NA NA  2  2 NA  0  1  2 NA  0  0  1  1 NA NA  1  0  0  0  2  1
## [47]  0  2 NA  0
##
## $`25`
##  [1]  1 NA  0  0 NA NA  0  2  0 NA NA  0  1  2  1  2 NA  0  0 NA  0  1  0
## [24]  0 NA NA  0  1  2 NA NA  2  0  2 NA  1  0  0 NA  0 NA  0  1  0 NA NA
## [47]  0  2 NA  0
##
## $`26`
##  [1]  1  0  2  2 NA  0 NA  1 NA  0  2  1  2  2 NA  0  2 NA  2  1  2 NA  2
## [24]  2  1  0 NA  0 NA NA NA  0  0  0  0  2  0 NA  1 NA  0  1 NA  0  0  0
## [47]  1  0 NA NA
##
## $`27`
##  [1]  0  0 NA  0  1  1  0 NA  2  2 NA  0  0 NA  1  0  2  0  1  2  1  0  2
## [24]  0  0  1 NA NA NA  0  0  2  2  2 NA  1  1  0  0  1  1 NA  0  0  1  2
## [47] NA  0  1  1
##
## $`28`
##  [1]  0  0  0  1 NA  0  1 NA  0  0  2  0 NA NA  0  2 NA  0  0 NA  0  1  0
## [24] NA  0  1  2 NA  1  1  0  1  2  0  0 NA  2  2 NA  2  0  2  2  1 NA  2
## [47] NA  0  2 NA
##
## $`29`
##  [1] NA  0 NA NA  0 NA NA  0  0  2 NA  2  2  2 NA  0 NA  0  2  0 NA  1 NA
## [24]  0 NA  0  0 NA  0 NA  0  1 NA  2  2 NA  2  1 NA NA NA  2 NA NA  1  0
## [47]  1 NA  0 NA
##
## $`30`
##  [1]  2 NA  0  0 NA  0  1 NA NA NA  1  2  0  1  2 NA  2  1  2  1 NA  0  0
## [24]  2  1 NA  2  2  0  0  0 NA NA  0  2 NA  0  0  0  2 NA NA NA  1  0 NA
## [47] NA  1  1  1
##
## $`31`
##  [1]  2 NA NA NA NA  1  1  2  0  2  1  0  0 NA  2 NA  2 NA  0 NA  0  0 NA
## [24]  1  0  0  0  0  0  1  0 NA  1  1 NA  0  0  2 NA NA  2  2 NA NA  1  1
## [47] NA  1  0  0
##
## $`32`
##  [1]  2  2  0 NA  1  0  1 NA NA  0  0  2  0  2 NA  0  2  2  0  1 NA  2  0
## [24] NA  0 NA  0  2  0 NA  0  0 NA NA NA  2 NA  0  0  0  0  2  0  2 NA NA
## [47] NA NA NA  0
##
## $`33`
##  [1]  0  2  1  0 NA  1  2  0  1 NA  0  0 NA  0  0  1  0  1  2 NA NA  2  0
## [24] NA  1  2  1 NA  0  2 NA NA  0 NA  0  1 NA NA NA  1 NA NA  0  1 NA NA
## [47]  0  2  0  1
##
## $`34`
##  [1]  0 NA  2 NA  2  1  0  2 NA  0  0  0  2  1  0  1  1  0  1  1  0  0  2
## [24] NA  0 NA NA NA NA  0  1 NA  2 NA NA  2  1  0  0  0 NA NA  2  0  2  0
## [47]  0  0 NA NA
##
## $`35`
##  [1]  2  1  2 NA  2  0  0  0  1  0 NA NA  1  1 NA NA  2  1 NA  0  1 NA NA
## [24]  0  0  0 NA NA NA NA  1  1  2  1  0 NA NA  0  1  0 NA NA NA NA  0  2
## [47]  0  0 NA  2
##
## $`36`
##  [1] NA  1  0  2  0  0 NA  0 NA  0  0  0  0 NA  0 NA  0  2  2  2  0  1 NA
## [24] NA NA  0  2 NA NA  2  1 NA  0 NA  2  0  2  1 NA  2  0 NA NA  0  1  0
## [47] NA  2  0  1
##
## $`37`
##  [1]  2  0  2  1  0  0 NA  0  0  0  0  1  0 NA  0 NA  0  1  2  2  0  0 NA
## [24]  1  1  0  1  1  0  0  1 NA  1  1  1  0  0  0 NA  0 NA  2  2  0  0 NA
## [47] NA  0  0 NA
##
## $`38`
##  [1]  2  0  1  0  0  1  1  1 NA NA  1  0  2  1  2  2  1 NA  2 NA NA NA  0
## [24]  1  1  0  0  0 NA  0 NA NA  2  1  1 NA  1 NA  0  0  0 NA  1  0  1  0
## [47]  2 NA  0  1
##
## $`39`
##  [1]  0  0 NA NA  0  0  1 NA  0  0  0  0 NA  2  0 NA  1 NA  1  1 NA  0 NA
## [24]  2  2  0 NA  0  0  0 NA  1  1 NA  2  0  0  2  0  2  0  2  2  2 NA  2
## [47]  1  2 NA  0
##
## $`40`
##  [1]  0  1 NA  2  0 NA  0 NA  2  1  1  1 NA  0  2  0  0  1 NA  2  0 NA  0
## [24]  2  1  0  1  2  2 NA  1  1 NA  1  2  0  0  0 NA NA NA  1  1  0 NA  2
## [47]  0 NA  0  0
##
## $`41`
##  [1] NA  0  0 NA NA NA  1  0  0  1  1 NA  0  2 NA NA NA  1 NA  0 NA NA  2
## [24]  0 NA  1  0 NA  2  2  0  1  2  2  0 NA  0  0  0 NA NA NA  1  2 NA  0
## [47]  1  0  2 NA
##
## $`42`
##  [1]  0  2  1  1 NA  0 NA  1  2  1  0 NA  0  2  0 NA  1  0  0  1 NA  0  1
## [24]  0 NA  0  0 NA  2 NA  0 NA NA NA  2 NA NA NA  0 NA NA NA  0 NA NA  2
## [47]  1  2  0  2
##
## $`43`
##  [1]  0  2  1  2  0 NA  0 NA  0 NA  1  1  0  0  0 NA  2  0 NA  0  0  0  0
## [24]  0  2 NA NA  0  2 NA  0  0  2 NA  0 NA  0  1 NA NA  2 NA  1  0  2  0
## [47]  0  0  0  1
##
## $`44`
##  [1] NA  0  1  1  1  1  0  0  1 NA  0 NA  1  0  0  0 NA NA  2  2  1  1  0
## [24]  2  2 NA  1 NA  2  2  0  0  0  1 NA  2 NA  2 NA  1  1  0  2  0 NA  1
## [47]  1  0  0 NA
##
## $`45`
##  [1]  1  2  2 NA  1  1  1  0 NA  0  0 NA  0 NA NA NA  1  0 NA  1 NA  0  0
## [24] NA  0  0 NA NA  0 NA NA  0 NA NA  2  0  1  0  1  0  1  0 NA NA NA NA
## [47]  0 NA  1  0
##
## $`46`
##  [1] NA  0  0 NA NA  2 NA  0  0  2  2  1  2  1 NA  0 NA  1 NA  2 NA  0  0
## [24] NA  1 NA  0 NA  2  2 NA  0 NA  0 NA  1  0  2 NA NA NA NA  2  0  0 NA
## [47] NA  0 NA  0
##
## $‘47‘
##  [1] NA  0  0 NA  2 NA  0  0  0  0 NA  0  2 NA  0 NA  0  0 NA  0 NA NA  1
## [24]  2  1  1  0  0 NA  2 NA NA NA  2  2 NA  1  0 NA  1  2  0  2  2  0  0
## [47] NA  2  0  0
##
## $‘48‘
##  [1] NA  0 NA  2 NA  1  0  2  0 NA  1 NA NA  1  1 NA NA  1 NA  0 NA  2 NA
## [24]  1  0  0  1  0  2  1  0 NA NA  1  1 NA  2  1  2 NA  2 NA  0  0 NA  1
## [47]  0 NA  1 NA
##
## $‘49‘
##  [1]  0  1  0  0  2  1 NA  0 NA NA  0  2 NA  0  1 NA  0  0  1  2  2  2 NA
## [24]  1  1  0  0  0 NA NA  0 NA  1 NA  1  0  0  2  2  0  2 NA NA  0  0  2
## [47] NA  1  0 NA
##
## $‘50‘
##  [1]  1  2  0  2  0 NA NA  2  0 NA  0 NA  0  2  0  0 NA  2  1 NA  0 NA  2
## [24]  0 NA  0  1  0  0  0 NA NA  0 NA  1 NA  2  0  0  0  0  0  2  0 NA NA
## [47] NA  2 NA  2
```

- 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(x){ c(min = min(x, na.rm = TRUE),
                                        max = max(x, na.rm=TRUE),
                                        pct_missing = sum(is.na(x))/length(x),
                                        first_NA = which.min(is.na(x)))})
```

```
## $‘1‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.24        1.00
##
## $‘2‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.26        1.00
##
## $‘3‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.28        1.00
##
## $‘4‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.26        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        0.16        1.00
##
## $`7`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.28        1.00
##
## $`8`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.32        1.00
##
## $`9`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.26        2.00
##
## $`10`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.32        1.00
##
## $`11`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.24        1.00
##
## $`12`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.36        2.00
##
## $`13`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.38        1.00
##
## $`14`
##         min         max pct_missing    first_NA
##         0.0         2.0         0.2         1.0
##
## $`15`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.28        1.00
##
## $`16`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.26        1.00
##
## $`17`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.18        1.00
##
## $`18`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.26        1.00
##
## $`19`
##         min         max pct_missing    first_NA
##        0.00        2.00        0.36        4.00
##
## $‘20‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.32        1.00
##
## $‘21‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.28        1.00
##
## $‘22‘
##         min         max pct_missing    first_NA
##         0.0         2.0         0.3         3.0
##
## $‘23‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.46        3.00
##
## $‘24‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.24        1.00
##
## $‘25‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.34        1.00
##
## $‘26‘
##         min         max pct_missing    first_NA
##         0.0         2.0         0.3         1.0
##
## $‘27‘
##         min         max pct_missing    first_NA
##         0.0         2.0         0.2         1.0
##
## $‘28‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.26        1.00
##
## $‘29‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.44        2.00
##
## $‘30‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.32        1.00
##
## $‘31‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.32        1.00
##
## $‘32‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.34        1.00
```

```
##
## $‘33‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.34        1.00
##
## $‘34‘
##         min         max pct_missing    first_NA
##         0.0         2.0         0.3         1.0
##
## $‘35‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.38        1.00
##
## $‘36‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.32        2.00
##
## $‘37‘
##         min         max pct_missing    first_NA
##         0.0         2.0         0.2         1.0
##
## $‘38‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.26        1.00
##
## $‘39‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.26        1.00
##
## $‘40‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.26        1.00
##
## $‘41‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.38        2.00
##
## $‘42‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.38        1.00
##
## $‘43‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.26        1.00
##
## $‘44‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.24        2.00
##
## $‘45‘
##         min         max pct_missing    first_NA
##         0.0         2.0         0.4         1.0
##
## $‘46‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.42        2.00
##
## $‘47‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.32        2.00
##
## $‘48‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.38        2.00
##
## $‘49‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.28        1.00
##
## $‘50‘
##         min         max pct_missing    first_NA
##        0.00        2.00        0.32        1.00
```

- 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.

```
set.seed(11)
v=rnorm(1000, mean=-10, sd=10)
head(v)
```

```
## [1] -15.9103110258  -9.7340563098 -25.1655309708 -23.6265334930
## [5]   1.7848915603 -19.3415131967
```

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

```
set.seed(11)
v=rnorm(1000, mean=-10, sd=10)
head(v)
```

```
## [1] -15.9103110258  -9.7340563098 -25.1655309708 -23.6265334930
## [5]   1.7848915603 -19.3415131967
```

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

```
mean(v)
```

```
## [1] -9.9120889267
```

```
SE = sd(v)/sqrt(1000)
SE
```

```
## [1] 0.3150349092
```

- 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?

Yes this is to be expected because v is made up of realization from norm(mean=-10, sd=10), and should be close to the 5th percentile of norm(mean=-10, sd=10).

```
quantile(v, probs = 0.05)
```

```
##             5%
## -26.366352937
```

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

```
## [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?

This is also to be expected by theory because v is made up of realizations from norm(mean=-10, sd=10), and so should have approximately the same CDF as norm(mean=-10, sd=10).

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

```
## [1] 0.845
```

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

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