Lab 1

Jordan Simkovic

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. Most of this will be a pure programming assignment but there are some questions that instead ask you to "write a few sentences". This is a W class! The tools for the solutions to these problems can be found in the class practice lectures. I prefer you to use the methods I taught you. If you google and find esoteric code you don't understand, this doesn't do you too much good.

To "hand in" the homework, you should first download this file. The best way to do this is by cloning the class repository then copying this file from the folder of that clone into the folder that is your personal class repository. Then do the assignment by filling in the TO-DO's. After you're done, compile this file into a PDF (use the "knit to PDF" button on the submenu above). This PDF will include output of your code. Then push the PDF and this Rmd file by the deadline to your github repository in a directory called "labs".

Basic R Skills

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Is this answer *exactly* correct?

#TO-DO

- 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] 0.53
```

- 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] 0.898
```

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

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

```
##    [1] 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1
##   [38] 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 1 1 0 1 1 1 1
##   [75] 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1
##  [112] 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 0 1 1 1 1 1 1
##  [149] 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 1 1
##  [223] 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1
##  [260] 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 1
##  [297] 1 1 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1
##  [334] 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 0 0 1 1 1 1 0 1 1
##  [371] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1
##  [408] 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
##  [445] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
##  [482] 0 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1
##  [519] 1 1 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [556] 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 1 1 0 0 1 1 1
##  [593] 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0
##  [630] 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 0 1 1 1 1 1 1 1 1 0 1
##  [667] 1 1 1 1 1 0 1 0 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
##  [704] 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
##  [741] 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 0 1 0 0 1 1 1 1
##  [778] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [815] 1 1 1 1 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
##  [852] 1 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 1 1 1 1 1 1 1 1 1 0 1 1 0 1
##  [889] 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 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
##  [926] 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1
##  [963] 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 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 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  misdimeanor misdimeanor none        infraction  felony
##   [7] infraction  misdimeanor misdimeanor none        misdimeanor misdimeanor
##  [13] none        infraction  none        infraction  none        felony
##  [19] misdimeanor felony      infraction  misdimeanor felony      none
##  [25] felony      infraction  felony      infraction  infraction  felony
##  [31] misdimeanor none        felony      felony      none        none
##  [37] felony      none        misdimeanor misdimeanor misdimeanor infraction
##  [43] infraction  infraction  felony      misdimeanor infraction  infraction
##  [49] none        felony      felony      misdimeanor none        misdimeanor
##  [55] misdimeanor felony      felony      none        misdimeanor felony
##  [61] misdimeanor infraction  infraction  none        felony      misdimeanor
##  [67] felony      misdimeanor infraction  felony      felony      infraction
##  [73] misdimeanor misdimeanor none        felony      infraction  infraction
##  [79] none        felony      none        none        infraction  none
##  [85] infraction  misdimeanor misdimeanor infraction  felony      none
##  [91] felony      infraction  infraction  infraction  felony      misdimeanor
##  [97] felony      infraction  none        none
## Levels: felony infraction misdimeanor none
```

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

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

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

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

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

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

```
##  [85] infraction  misdimeanor misdimeanor infraction  felony      none
##  [91] felony      infraction  infraction  infraction  felony      misdimeanor
##  [97] felony      infraction  none        none
## Levels: none < infraction < misdimeanor < felony
```

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

```
y_1 = rep(0, length(x_3))
y_2 = rep(0, length(x_3))
y_3 = rep(0, length(x_3))

for (i in c(1:length(x_3))) {
  if (x_3[i] == "infraction") {
    y_1[i] = 1
  } else if (x_3[i] == "misdimeanor") {
    y_2[i] = 1
  } else if (x_3[i] == "felony") {
    y_3[i] = 1
  }
}

m = cbind(y_1, y_2, y_3)

m
```

```
##        y_1 y_2 y_3
##   [1,]   1   0   0
##   [2,]   0   1   0
##   [3,]   0   1   0
##   [4,]   0   0   0
##   [5,]   1   0   0
##   [6,]   0   0   1
##   [7,]   1   0   0
##   [8,]   0   1   0
##   [9,]   0   1   0
##  [10,]   0   0   0
##  [11,]   0   1   0
##  [12,]   0   1   0
##  [13,]   0   0   0
##  [14,]   1   0   0
##  [15,]   0   0   0
##  [16,]   1   0   0
##  [17,]   0   0   0
##  [18,]   0   0   1
##  [19,]   0   1   0
##  [20,]   0   0   1
##  [21,]   1   0   0
##  [22,]   0   1   0
##  [23,]   0   0   1
##  [24,]   0   0   0
##  [25,]   0   0   1
##  [26,]   1   0   0
##  [27,]   0   0   1
##  [28,]   1   0   0
```

```
##  [29,]   1   0   0
##  [30,]   0   0   1
##  [31,]   0   1   0
##  [32,]   0   0   0
##  [33,]   0   0   1
##  [34,]   0   0   1
##  [35,]   0   0   0
##  [36,]   0   0   0
##  [37,]   0   0   1
##  [38,]   0   0   0
##  [39,]   0   1   0
##  [40,]   0   1   0
##  [41,]   0   1   0
##  [42,]   1   0   0
##  [43,]   1   0   0
##  [44,]   1   0   0
##  [45,]   0   0   1
##  [46,]   0   1   0
##  [47,]   1   0   0
##  [48,]   1   0   0
##  [49,]   0   0   0
##  [50,]   0   0   1
##  [51,]   0   0   1
##  [52,]   0   1   0
##  [53,]   0   0   0
##  [54,]   0   1   0
##  [55,]   0   1   0
##  [56,]   0   0   1
##  [57,]   0   0   1
##  [58,]   0   0   0
##  [59,]   0   1   0
##  [60,]   0   0   1
##  [61,]   0   1   0
##  [62,]   1   0   0
##  [63,]   1   0   0
##  [64,]   0   0   0
##  [65,]   0   0   1
##  [66,]   0   1   0
##  [67,]   0   0   1
##  [68,]   0   1   0
##  [69,]   1   0   0
##  [70,]   0   0   1
##  [71,]   0   0   1
##  [72,]   1   0   0
##  [73,]   0   1   0
##  [74,]   0   1   0
##  [75,]   0   0   0
##  [76,]   0   0   1
##  [77,]   1   0   0
##  [78,]   1   0   0
##  [79,]   0   0   0
##  [80,]   0   0   1
##  [81,]   0   0   0
##  [82,]   0   0   0
```

```
##  [83,]   1   0   0
##  [84,]   0   0   0
##  [85,]   1   0   0
##  [86,]   0   1   0
##  [87,]   0   1   0
##  [88,]   1   0   0
##  [89,]   0   0   1
##  [90,]   0   0   0
##  [91,]   0   0   1
##  [92,]   1   0   0
##  [93,]   1   0   0
##  [94,]   1   0   0
##  [95,]   0   0   1
##  [96,]   0   1   0
##  [97,]   0   0   1
##  [98,]   1   0   0
##  [99,]   0   0   0
## [100,]   0   0   0
```

```
print(m)
```

```
##        y_1 y_2 y_3
##   [1,]   1   0   0
##   [2,]   0   1   0
##   [3,]   0   1   0
##   [4,]   0   0   0
##   [5,]   1   0   0
##   [6,]   0   0   1
##   [7,]   1   0   0
##   [8,]   0   1   0
##   [9,]   0   1   0
##  [10,]   0   0   0
##  [11,]   0   1   0
##  [12,]   0   1   0
##  [13,]   0   0   0
##  [14,]   1   0   0
##  [15,]   0   0   0
##  [16,]   1   0   0
##  [17,]   0   0   0
##  [18,]   0   0   1
##  [19,]   0   1   0
##  [20,]   0   0   1
##  [21,]   1   0   0
##  [22,]   0   1   0
##  [23,]   0   0   1
##  [24,]   0   0   0
##  [25,]   0   0   1
##  [26,]   1   0   0
##  [27,]   0   0   1
##  [28,]   1   0   0
##  [29,]   1   0   0
##  [30,]   0   0   1
##  [31,]   0   1   0
##  [32,]   0   0   0
##  [33,]   0   0   1
```

```
##  [34,]   0   0   1
##  [35,]   0   0   0
##  [36,]   0   0   0
##  [37,]   0   0   1
##  [38,]   0   0   0
##  [39,]   0   1   0
##  [40,]   0   1   0
##  [41,]   0   1   0
##  [42,]   1   0   0
##  [43,]   1   0   0
##  [44,]   1   0   0
##  [45,]   0   0   1
##  [46,]   0   1   0
##  [47,]   1   0   0
##  [48,]   1   0   0
##  [49,]   0   0   0
##  [50,]   0   0   1
##  [51,]   0   0   1
##  [52,]   0   1   0
##  [53,]   0   0   0
##  [54,]   0   1   0
##  [55,]   0   1   0
##  [56,]   0   0   1
##  [57,]   0   0   1
##  [58,]   0   0   0
##  [59,]   0   1   0
##  [60,]   0   0   1
##  [61,]   0   1   0
##  [62,]   1   0   0
##  [63,]   1   0   0
##  [64,]   0   0   0
##  [65,]   0   0   1
##  [66,]   0   1   0
##  [67,]   0   0   1
##  [68,]   0   1   0
##  [69,]   1   0   0
##  [70,]   0   0   1
##  [71,]   0   0   1
##  [72,]   1   0   0
##  [73,]   0   1   0
##  [74,]   0   1   0
##  [75,]   0   0   0
##  [76,]   0   0   1
##  [77,]   1   0   0
##  [78,]   1   0   0
##  [79,]   0   0   0
##  [80,]   0   0   1
##  [81,]   0   0   0
##  [82,]   0   0   0
##  [83,]   1   0   0
##  [84,]   0   0   0
##  [85,]   1   0   0
##  [86,]   0   1   0
##  [87,]   0   1   0
```

```
##  [88,]   1   0   0
##  [89,]   0   0   1
##  [90,]   0   0   0
##  [91,]   0   0   1
##  [92,]   1   0   0
##  [93,]   1   0   0
##  [94,]   1   0   0
##  [95,]   0   0   1
##  [96,]   0   1   0
##  [97,]   0   0   1
##  [98,]   1   0   0
##  [99,]   0   0   0
## [100,]   0   0   0
```

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

either 0 (if the person represented by the current row did not commit a crime) or 1 (if they did)

Verify that.

```
rows = rowSums(m)

identical(sort(unique(rows)), sort(c(0,1)))
```

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

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

It should tell you the total number of people who commited a infraction, misdimeanor or felony, respectively

Verify that.

```
column = colSums(m)
csum = sum(column)

csum == sum(rows == 1)
```

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

- 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"
)

norm = rnorm(100, 17, 38)
uni = runif(100, min = -10, max = 10)
pois = rpois(100, lambda = 6)
exp = rexp(100, rate = 1/9)
bin = rbinom(n = 100, size = 20, prob = 0.12)
bern = c(seq(from = 1, to = 100, size = 100))
```

```
## Warning: In seq.default(from = 1, to = 100, size = 100) :
##  extra argument 'size' will be disregarded
```

```
sample_vec = sample(c(1:100), size = 100, replace = FALSE)
j = 0
for (i in sample_vec) {
  if (j < 24)
    bern[i] = 1
   else
    bern[i] = 0
  j = j + 1
}

randm = cbind(norm, uni, pois, exp, bin, bern)
rownames(randm) = fake_first_names
```

- 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.

```
D_F = bern
D_F[D_F == 0] = "DOMESTIC"
D_F[D_F == 1] = "FOREIGN"

df = data.frame(normal = norm, uniform = uni, poison = pois, exponential = exp,
                binomial = bin, "D v F" = factor(D_F, c("DOMESTIC", "FOREIGN")))
View(df, "My data frame")
```

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

```
table(df[, 6])
```

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

```
prop.table(table(df[, 6]))
```

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

Print out a summary of the whole dataframe.

```
summary(df)
```

```
##      normal              uniform               poison
```

```
##  Min.   :-116.040818   Min.   :-9.44261639   Min.   : 1.00
##  1st Qu.: -10.361375   1st Qu.:-4.40217080   1st Qu.: 4.00
##  Median :  18.139526   Median : 1.57047377   Median : 6.00
##  Mean   :  16.068062   Mean   : 0.89818166   Mean   : 5.96
##  3rd Qu.:  45.778669   3rd Qu.: 5.99647279   3rd Qu.: 8.00
##  Max.   : 121.204667   Max.   : 9.93002751   Max.   :15.00
##   exponential              binomial         D.v.F
##  Min.   : 0.022722858   Min.   :0.00   DOMESTIC:76
##  1st Qu.: 2.160988096   1st Qu.:1.00   FOREIGN :24
##  Median : 5.876758990   Median :2.00
##  Mean   : 8.420547262   Mean   :2.46
##  3rd Qu.:11.369273821   3rd Qu.:3.00
##  Max.   :44.976894863   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.

```
mat = matrix(5, nrow = 50, ncol = 50)

x = sample(1:length(mat),round(length(mat)*.5))



y = setdiff(sample(1:length(mat),length(mat)), x)



mat[x] = 0

table(mat)
```

```
## mat
##    0    5
## 1250 1250
```

```
z = sample(y,round(length(mat)*.25))
mat[z] = 1

table(mat)
```

```
## mat
##    0    1    5
## 1250  625  625
```

```
w = setdiff(sample(1:length(mat),length(mat)), union(x,z))
mat[w] = 2

table(mat)
```

```
## mat
##    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%.

```
mat[sample(1:length(mat),round(length(mat)*.3))] = NA
```

```
table(mat)
```

```
## mat
##   0   1   2
## 872 438 440
```

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

```
mat[order(rowSums(mat, na.rm = TRUE),decreasing=T),]
```

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

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

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

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

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

- 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(mat, 2, sd, na.rm = TRUE)
```

```
##  [1] 0.76986467779 0.81523946458 0.88330492811 0.76041839198 0.73192505471
##  [6] 0.95442357984 0.86645874152 0.80361912523 0.76635604473 0.80950789391
## [11] 0.78532422798 0.76041839198 0.85449325928 0.91919518390 0.77024496813
## [16] 0.82615959871 0.87423435890 0.82836355919 0.76481777997 0.86383570267
## [21] 0.80445456500 0.82733053035 0.86772183127 0.81167944991 0.67891055392
## [26] 0.88991798666 0.72081075959 0.83313723183 0.78978182701 0.86309864515
## [31] 0.75037528148 0.77408420033 0.80229046222 0.83971912276 0.92309308325
## [36] 0.78857386432 0.83390784794 0.87255059327 0.87128628185 0.93338744432
## [41] 0.75996059566 0.77401492175 0.88611864831 0.89306822259 0.87038827978
## [46] 0.90632696717 0.87240113700 0.90792308282 0.78313504111 0.85901293693
```

- 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(mat == 1, 2, sum, na.rm = TRUE) + apply(mat == 2, 2, sum, na.rm = TRUE)
```

```
##  [1] 20 14 18 15 16 16 20 25 16 16 16 15 21 19 20 25 21 15 13 20 19 14 18 14 19
## [26] 11 15 19 16 17 22 13 19 20 17 18 17 23 13 20 18 24 20 16 18 20 14 18 11 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(mat, col(mat))
```

```
## $`1`
##  [1]  0  1 NA  0  2  1  1  1  0  0  2 NA NA NA  0  2  0  0  1  0 NA  1  1  2  2
## [26]  1  1 NA NA NA  0 NA  0  0 NA  1  2  1  0  0 NA NA NA  0  2  1 NA NA  1 NA
##
## $`2`
##  [1]  0  0  1 NA NA  0 NA NA NA  0  1  2 NA  1  2 NA  2 NA NA NA  1  0  2 NA NA
## [26]  0  0  1 NA NA NA  1  0  0  1 NA NA NA  1  2 NA NA NA NA NA  0 NA  0  0  2
##
## $`3`
##  [1]  2 NA NA  1  0  0  0  1 NA  1  0  1  0 NA  2 NA  0  2  0  0 NA  0  2  2 NA
## [26]  2  2  1 NA  0  0  0  0  0  0  2  0  0 NA  2 NA NA  2  2  0 NA  1  0  0  1
##
## $`4`
##  [1]  0  2 NA  0  0  1  0  1  1  0  0  0  1 NA  0  2  0  1 NA NA  0 NA  1  0  0
## [26]  0  2  1 NA  2  2  2 NA NA  0  0  0 NA  1  0  0  0 NA  0 NA  1  0 NA  0  0
##
## $`5`
##  [1]  2  0 NA  0  2 NA  1  0  0  1  0  1  0 NA NA  1 NA  1 NA NA NA  0 NA  2  0
## [26]  2  0 NA  1  1  0  0  0  1  1  0  1 NA  0  1 NA  0  2  0  0  0 NA  0 NA  0
##
## $`6`
##  [1]  0  0  0 NA NA  2  0  0 NA  0  2  0  0  1 NA  2  2  0 NA  0  2 NA  0  2  0
## [26]  0  0  0 NA NA  1  2  0 NA  0 NA NA  2  2 NA  2  1  2  0 NA NA  2  0 NA  2
##
## $`7`
##  [1]  1  2  2  0 NA  1  2 NA  0  1  0 NA  0 NA  0  1 NA  0  2  0 NA NA  1  1  0
## [26]  0  0 NA  2  2  1  0 NA  1  2  0 NA  0  2  0  1  2  2 NA NA  0 NA  0  2  0
##
## $`8`
##  [1]  2  0  2 NA  0  0  1  1  1  0  2  0  1  2  1  2  2  0  2  0  0  0  1  1  2
```

```
## [26]  0  1 NA  0 NA NA  2 NA NA NA  2 NA  1  0  1  1  1 NA NA  1  0 NA  1  2 NA
##
## $`9`
##  [1]  1  0  0  0  0  2  2  0  1  1 NA  0 NA NA  2 NA  0  2 NA NA  0 NA  1 NA  0
## [26]  0  1 NA  1  0 NA  0  0  1 NA  0  2  0 NA  0  2 NA  0  0  1  1  1 NA  0  0
##
## $`10`
##  [1] NA  2  2 NA  1  2  0 NA NA  0  2 NA  2 NA  0  0  0 NA  1 NA  1  0  1 NA NA
## [26]  0  1  0  0  2  0  1 NA NA  0  0 NA  0  1  0  0  1 NA  0  0  1 NA  2 NA NA
##
## $`11`
##  [1]  0  0  2  0  0  0 NA  0  0  1  0 NA NA  0  1  1  2  0  0  1  0  2  1 NA  0
## [26] NA NA  0  1  0  2  1  0 NA  1  0 NA  0  0 NA  0  0  2  0  2 NA  2 NA  1  0
##
## $`12`
##  [1]  1  0  0  0  0  0  0  1 NA NA NA  0 NA  1  0  0  0  0  2  1  0  1  2 NA  0
## [26] NA NA  0  1  2  2  0  1 NA NA  0  0  0 NA  0 NA  2 NA  0  1  0  2  1  0  0
##
## $`13`
##  [1]  1  1  2 NA  1  2 NA  0  0  0  1  2  2 NA  1  2  0  0 NA  2  0  0  1  2  1
## [26]  2  0 NA NA  2  2  0 NA NA NA NA  2  1  1 NA  0  0  0 NA  0 NA  0  1  0 NA
##
## $`14`
##  [1]  0  2  2  0  0 NA  1 NA  0  2  2 NA NA  2  2  0 NA  1  0  2  0 NA  0  2 NA
## [26] NA  1  0 NA NA  1 NA NA  0  2  0  1 NA  2 NA NA NA  1  0  0  0  2  2  0  2
##
## $`15`
##  [1]  0 NA  1 NA  0  2  1 NA  0 NA  0  1 NA  0  1  1  1  0  0  2 NA NA  0 NA NA
## [26]  0  1  0  1  1  0  1  0 NA  2  2  0  0 NA  0 NA  1  2  2  2  1 NA  1 NA NA
##
## $`16`
##  [1]  0  1 NA NA NA  1  2  1  1  2  2  0  2  1  0 NA  2  1  1  2 NA  0  0 NA  0
## [26]  2  1  1  2 NA NA NA NA  1  2  2  1  0  2  0 NA  1  2  0  0 NA  0 NA NA  2
##
## $`17`
##  [1]  0  0  0  0  2  1  0  2  0  1  1 NA NA  0  1  1  1  2  0  1 NA  0  1 NA  2
## [26] NA NA  2 NA  2  2  0  2  2  2 NA  2  0  1 NA NA  0  2 NA  0 NA NA  0  0 NA
##
## $`18`
##  [1]  0 NA  2  0  0 NA  1  0  0  0 NA  1 NA  2  0 NA  1  2  0  0  0 NA  0  2  0
## [26]  1  2 NA  0  0  0 NA NA NA  0 NA  2  1  1  2  1  0 NA  0 NA  0  0  0  0  2
##
## $`19`
##  [1]  1  1 NA  0  1 NA  0  0  2  0 NA  1  0  1  0 NA NA  1  0  0 NA  2 NA  2  0
## [26] NA NA  0 NA NA  0  2 NA  0 NA  0 NA  1  2  0 NA  0  1  0 NA  0 NA NA  0 NA
##
## $`20`
##  [1]  1  1 NA  0 NA NA NA  1 NA NA  2  2  1  2  2  2  0  0  0  2 NA  2  1 NA NA
## [26]  2  2  0  0  0 NA  1  2 NA  0  2 NA  0 NA  0  1  0  1  0 NA  1 NA NA  0 NA
##
## $`21`
##  [1]  1  0  0  1  0  0  0  1  1  1  2  0  1 NA  0  1  1  2  0 NA  0 NA  0  1 NA
## [26] NA  1 NA  2  2  0  0  0  1  2  0  0  0 NA NA  2 NA  2  0 NA NA  0 NA NA  2
##
```

```
## $`22`
##  [1]  2 NA  1  1 NA  1  0 NA  2  0  0  2  1 NA NA  0 NA  0  2  0  0 NA  1  0  0
## [26] NA  2  0  2  0  0 NA NA NA NA NA  0  0 NA NA  0 NA  1 NA  1  2  0  0  0 NA
##
## $`23`
##  [1] NA NA NA  2  1  0  2  2  0  0  2 NA  0  2 NA  0 NA  0  0 NA  1  0  2  1  1
## [26]  0 NA  0  0 NA  1 NA  0  0  2  1 NA  2  2  1  0  0 NA  1 NA NA  2 NA  0  0
##
## $`24`
##  [1]  0  0  0 NA  0  2 NA  1  0  2  0  0  0  0 NA  0  0 NA  2 NA NA  0  0 NA  2
## [26]  1 NA NA  1  0 NA NA  1  1  0  2 NA  0  0  2  2  0  0 NA  1 NA  0  0  1 NA
##
## $`25`
##  [1] NA  1  0  1 NA  1  1 NA  0  0 NA NA  0  1 NA NA NA  1 NA  0  2  2  1  1 NA
## [26]  0  2 NA  0  1  2  1  1  0 NA  1 NA  0  1 NA  1  0 NA NA NA  0 NA NA  1 NA
##
## $`26`
##  [1] NA  2  0  0  0 NA NA NA  1 NA  0  0 NA NA  0  2  0  0 NA NA  2  1  0  0  2
## [26]  1  0 NA NA NA NA  0  2 NA  2  0 NA  0  0 NA  0 NA  0 NA  0  2 NA  2 NA  0
##
## $`27`
##  [1]  0  1 NA  0  1  0  1  2  0  0  0  1  0  0  0 NA  0  0  0 NA  1  0  0  2 NA
## [26] NA  0  1  2 NA  0 NA  0  1  0  0  1  0  1 NA  0  2 NA NA  0  0  1 NA  2  0
##
## $`28`
##  [1]  1  1  0  2  2 NA NA  2  2  0  0  2  0 NA  0  0 NA  0  0 NA  0  1  0 NA  1
## [26] NA  2  0  0 NA NA  0  1  0 NA NA  2  1  1  1  2  0 NA  1 NA  2  0  1 NA NA
##
## $`29`
##  [1]  0  2  0 NA NA  1  0  0 NA  1 NA NA  0  0  2  0  0  1  0 NA  1  0  0  0  0
## [26]  1  0 NA  2  0  2 NA  1  1  0  2  1  0 NA NA  0  0 NA  0  0 NA  0  1  2  2
##
## $`30`
##  [1] NA  0  2 NA  2  0  0  0  0  0  0  0  1  0 NA NA  0  2  0  2  0  1  2  0  2
## [26]  1  1  2 NA  2 NA  0 NA  0  0  2  1 NA  0  0 NA  0  0  0  2  1  1 NA NA  0
##
## $`31`
##  [1] NA NA  1 NA NA NA  0  0  0  1  2  0 NA NA  0  1  2  0  1  0  2  2 NA  0 NA
## [26]  1  1 NA NA  0  1  1  0  1  2  0  1  0  2  1  2  1 NA  1  0 NA  1  0  0  1
##
## $`32`
##  [1]  0  0  0 NA  0 NA  0 NA  1  0  0  0  0  2  0  1  0  1 NA  1 NA NA  0 NA  0
## [26]  1  0 NA NA NA  1  0  0 NA  2  0  2  0 NA NA  1 NA  0  2  0  2  0  0  0  2
##
## $`33`
##  [1]  1 NA  0  0  0  0  1  0  1 NA  1 NA  0  0  0  1 NA  2  1  1  2  2  0  0  0
## [26]  0  2  1 NA  1 NA NA NA  0 NA  0  1  0  0  2 NA NA  2 NA  0  2  0  2  1  0
##
## $`34`
##  [1]  0  2  2 NA  0 NA  2  2 NA  1  1  1  0  1  1  0  2  0  0  1  0  0  0  2  0
## [26]  1  1  2  0  0  2  1  0 NA  2  0  0 NA NA NA  0 NA  2 NA  0  0 NA  0  0  1
##
## $`35`
##  [1]  2 NA  0  0  0  0 NA NA  0  0  2  0 NA  0  0  1 NA  0  0  2  2  2 NA  1 NA
```

```
## [26] NA  1 NA NA  2 NA NA NA  2 NA  2  0  1 NA  0  2  0  0  2  2  0  0  2  1  0
##
## $`36`
##  [1]  0 NA NA  0 NA  2  1 NA  1  1  0 NA NA NA NA  2  1  0  0  0  0 NA  2 NA  2
## [26]  0  0 NA  2  0  0  1  1  1 NA  0  0  1  0  1 NA NA  2  1  2 NA  0  0  0  1
##
## $`37`
##  [1] NA NA  1 NA  2 NA  0  0  1  0  1  0 NA  1  0  1  0  0  0 NA NA NA  0 NA  2
## [26]  2 NA  0 NA NA NA NA  2  1  2  0 NA  1  2 NA NA NA  2  1  0  1  2 NA  0 NA
##
## $`38`
##  [1]  1  2  1 NA  0  1  0  2  0  1  2  0  1 NA NA  0 NA  2  1  0 NA NA  0  0  0
## [26] NA  2 NA NA  1  2  0  2  0 NA NA  2 NA  1  2  2  2 NA  1  2  2  0 NA  2 NA
##
## $`39`
##  [1] NA NA  0  2 NA  2  2  0  0  0 NA NA  0  0 NA  0  0  0  0  2 NA  2  1 NA  1
## [26]  0  0  2 NA NA  0  1 NA NA NA  1  2  0  0 NA NA NA  1  0  0  0 NA NA NA  2
##
## $`40`
##  [1]  0  2  0  2 NA NA  2  0  0  2  0 NA  1 NA NA NA  2  2 NA  2  0  1  0  2  2
## [26]  2 NA  0  2  1 NA  1  2 NA  0  0  2 NA  0 NA  1 NA  0  2 NA NA NA NA  0  2
##
## $`41`
##  [1]  2 NA  0  0 NA  1  0  1  1 NA  1 NA  2 NA NA NA NA  2  2  0  1  0  0  0  0
## [26] NA  0  0  0 NA  0 NA  0  2 NA  0  1  1 NA  0  0  1 NA  1 NA  2  1 NA  1  1
##
## $`42`
##  [1]  2 NA  0 NA  1  2  2  1 NA NA  2  2 NA  0  2  1  1  1  0  0  0  1  2  1 NA
## [26] NA NA  1  0  0  1  2  1 NA NA NA  0  0  1  2  1  1  1  1  0  0 NA NA  0  0
##
## $`43`
##  [1]  0  1 NA  2 NA  2  2  1 NA NA  0  2  2 NA  1  0 NA  2  1  1  0  1 NA  1  0
## [26] NA  1  2  2  0 NA  0 NA  2  2 NA  2  0  0  0  0  0  0  0 NA  0  0  2  0  0
##
## $`44`
##  [1]  0  2  0  0  2  0  0  0  0  0  2  0  2  2  2  1  2  0 NA NA  0  0  0  1  0
## [26] NA NA NA  0  1  1  2  0 NA  0 NA  2  0  0 NA  0  1  2 NA  0 NA  2  0  0 NA
##
## $`45`
##  [1]  0 NA  1  0 NA  1  0  2 NA NA NA  0  2  0  0 NA  0  0  1  2  1 NA  1  0  0
## [26]  2  1  2 NA  0  2  0 NA  2 NA  1  2  2 NA NA  0 NA  0 NA NA NA NA  2  1  0
##
## $`46`
##  [1]  0  2  0  0  0  2  1 NA  0 NA  1  2  1 NA  0  1 NA NA  2  0  2  2  1  2  2
## [26]  2  0 NA  0  0  0  0 NA NA  0 NA NA  2  2  1 NA NA NA  2  2 NA  0  0  1  0
##
## $`47`
##  [1]  1 NA NA  2  2 NA  1 NA  0  0 NA NA  2  0  0  0  0  0 NA  0 NA NA  0  2 NA
## [26] NA  0  0  2 NA NA  0 NA NA  2  0 NA NA  1  0  1 NA  2 NA  2  1  0  1 NA NA
##
## $`48`
##  [1]  2  0  0  0  2  2  2 NA NA  2 NA  0 NA  0  1  0 NA NA  0 NA NA  0 NA  0  2
## [26]  1  2  2 NA NA  0  1 NA  0  0  0  1  2  0  0 NA  2  0  0  2  1  0  0  1  2
##
```

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

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

```
#lapply(col(mat))
```

- 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(1984)
v = rnorm(1000, -10, 100)
```

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

```
set.seed(1984)
r = rnorm(1000, -10, 100)

all.equal(v,r)
```

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

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

```
mean(v)
```

```
## [1] -14.033373206
```

```
sd(v)
```

```
## [1] 99.026738881
```

- 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?

```
comp = quantile(v, .05)

theo = qnorm(.05, -10, 100)

all.equal(comp, theo)
```

```
## [1] "names for target but not for current"
## [2] "Mean relative difference: 0.0075606036773"
```

- 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?

```
comp = ecdf(v)(0)

theo = pnorm(0, -10, 100)

all.equal(comp, theo)
```

```
## [1] "Mean relative difference: 0.029086623602"
```