Lab 2

Tyron Samaroo

11:59PM February 17, 2020

More Basic R Skills

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

```
mean(rbinom(100, size = 1, prob =  0.9))
```

```
## [1] 0.89
```

- In class we considered a variable `x_3` which measured "criminality". We imagined L = 4 levels "none", "infraction", "misdimeanor" and "felony". Create a variable `x3` here with 100 random elements (equally probable). Create it as a nominal (i.e. unordered) factor.

```
x_3= sample(c("none", "infraction", "misdimeanor", "felony"),
            size = 100,
            replace = TRUE)
#x_3_ord = factor(sample_criminality, levels = c("none", "infraction", "misdimeanor", "felony"), ordere
x_3
```

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

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

```
X = matrix(nrow= length(x_3), ncol = 3)
X[ ,1] = as.numeric(x_3 == 'infraction')
X[ ,2] = as.numeric(x_3 == 'felony')
```

```
X[ ,3] = as.numeric(x_3 == 'misdimeanor')
colnames(X) = c('is_infraction', 'is_felony', 'is_misdimeanor')
X
```

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

```
##  [50,]            0         0            0
##  [51,]            0         1            0
##  [52,]            0         1            0
##  [53,]            0         0            1
##  [54,]            0         1            0
##  [55,]            0         0            0
##  [56,]            0         0            0
##  [57,]            0         0            0
##  [58,]            0         1            0
##  [59,]            0         0            1
##  [60,]            0         0            1
##  [61,]            0         1            0
##  [62,]            0         1            0
##  [63,]            0         1            0
##  [64,]            0         0            1
##  [65,]            0         0            1
##  [66,]            0         0            1
##  [67,]            0         1            0
##  [68,]            0         0            1
##  [69,]            0         0            0
##  [70,]            0         1            0
##  [71,]            0         0            0
##  [72,]            0         1            0
##  [73,]            0         0            0
##  [74,]            1         0            0
##  [75,]            0         1            0
##  [76,]            0         0            1
##  [77,]            0         0            0
##  [78,]            1         0            0
##  [79,]            1         0            0
##  [80,]            0         0            0
##  [81,]            1         0            0
##  [82,]            1         0            0
##  [83,]            1         0            0
##  [84,]            0         0            0
##  [85,]            0         0            1
##  [86,]            1         0            0
##  [87,]            0         1            0
##  [88,]            0         0            1
##  [89,]            0         0            0
##  [90,]            1         0            0
##  [91,]            0         0            0
##  [92,]            0         0            0
##  [93,]            0         0            1
##  [94,]            0         0            1
##  [95,]            1         0            0
##  [96,]            1         0            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)? Verify that. It should be either 1 or 0 because categories are mutally exclusive. We can only put an object into 1 category, None = 0

```
rowSums(X)
```

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

```
table(rowSums(X))
```

```
##
##  0  1
## 27 73
```

- How should the column sum look (in English)? Verify that. The should should be around the expectation 25 since they are uniforml distructed

```
colSums(X)
```

```
##  is_infraction      is_felony is_misdimeanor
##             31             19             23
```

- 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 columns based on the r.v. Name the rows the entries of the `fake_first_names` vector.

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


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"
)
rownames(X) = fake_first_names
X
```

```
##                [,1]       [,2] [,3]        [,4] [,5] [,6]
## Sophia    17.8460639 -9.5060941    5 0.038361642    2    0
## Emma      28.4915007 -8.9494042    6 0.267481023    3    0
## Olivia    16.5854290 -6.7571072    4 0.320759563    2    1
## Ava       13.6426384  3.2624797   10 0.056956998    2    0
## Mia       11.1561830 -1.0958796    7 0.033177328    1    1
## Isabella   9.2997628 -5.2515940    6 0.149927116    1    0
## Riley     22.9104565  6.1770309    8 0.164080251    6    1
## Aria      11.1231149  9.3543350   11 0.185602755    2    0
## Zoe       11.9161521  3.2380188    4 0.183038683    1    0
## Charlotte  7.7076312  1.2894057    9 0.126249892    2    0
## Lily      17.2432819  6.1564785    2 0.008464963    2    0
## Layla     15.0086994 -1.2397703    4 0.452499452    2    0
## Amelia     3.3679293 -1.8651010    7 0.108566045    4    0
## Emily     16.1034771 -8.1924464    5 0.298669357    5    0
## Madelyn   18.3381568  7.2429860    4 0.106924399    3    0
## Aubrey    18.8155499  3.5943590    6 0.382279928    4    0
## Adalyn    24.7798129  6.7852624    9 0.057037102    2    0
## Madison   16.5070152 -2.4699173    4 0.003585211    4    1
## Chloe     10.4706402  8.8440410    4 0.165242959    0    1
## Harper    26.7656433 -8.4924086    4 0.007587857    3    0
## Abigail   11.8064633 -1.2552792   11 0.085680424    3    0
## Aaliyah   11.5997922  2.6650190    3 0.134880841    3    1
## Avery     17.1584074 -4.3875140    4 0.104212531    2    1
## Evelyn    21.2929363  3.6682151    6 0.167579909    3    0
## Kaylee    18.2358792 -6.0737976    3 0.126087118    2    0
## Ella      15.0437353 -2.0431593    7 0.108505565    1    0
## Ellie     22.3032292 -3.5904466    4 0.216496446    0    0
## Scarlett   7.4616564  8.5137227    9 0.187972426    1    1
## Arianna   15.9293528 -9.9830577    4 0.006363437    3    0
## Hailey    19.5855909  9.2276552    8 0.102149615    1    0
## Nora      20.2532767  4.9183549    9 0.059694295    3    1
## Addison   33.3456553  1.5298770    6 0.020926553    3    0
## Brooklyn  12.2121592 -7.6243079    2 0.105940416    3    0
## Hannah     0.7551125 -1.5447252    7 0.144820232    3    0
## Mila      14.9503566 -4.3306047    5 0.250470975    2    0
## Leah       9.0576792  9.7354602    8 0.007902291    4    0
## Elizabeth 14.9684417  2.7669378    8 0.082905491    4    0
## Sarah     13.1405139 -4.0523690    4 0.025734279    2    1
## Eliana    12.3896418  1.4772260    5 0.141967242    2    0
## Mackenzie 20.3705252  5.1017751    5 0.080281952    3    0
## Peyton     6.5280722  5.2138714    5 0.003543715    2    0
## Maria      1.7394001 -7.3650968    4 0.012500477    6    0
## Grace      7.1753630  1.2419093    6 0.006903284    0    0
## Adeline   19.4658398 -5.1258283    6 0.090066574    2    0
## Elena     15.4285953  9.4307998    8 0.475183383    3    0
## Anna       7.3502659 -1.7272058    4 0.032556350    3    0
## Victoria  11.0452077 -9.1667514    7 0.114661936    0    0
## Camilla   21.3039494 -8.7682942    5 0.568942244    3    0
## Lillian   21.8340344  9.3304956    6 0.001679693    3    0
## Natalie   10.6438132  3.6276964    2 0.005753041    4    0
## Jackson   27.8076653 -0.1086385    2 0.127087859    2    0
## Aiden     24.5353613 -8.8936180    3 0.207745144    3    1
## Lucas     12.2851798  8.9568441   13 0.222767339    0    1
```

```
## Liam      19.6230118  1.5540945   12 0.303399984    3    0
## Noah      25.8673359  5.7168356    9 0.281435189    0    0
## Ethan     19.6126825  4.7062299    4 0.052532391    2    1
## Mason     15.4016300 -6.4346912    8 0.065530862    5    0
## Caden     13.7905921  7.9200834    6 0.268540250    1    0
## Oliver     7.8451608 -2.7757978    8 0.026458621    4    0
## Elijah    17.1831069  9.6767045    5 0.020582555    3    1
## Grayson   18.4184724 -6.3467487    6 0.037672199    3    0
## Jacob     26.1388981  5.0119568    7 0.003480217    1    0
## Michael   26.2660677  6.0427397    8 0.042675469    0    0
## Benjamin  22.0588916  7.7028747    5 0.230381360    5    0
## Carter    23.0800879 -9.4443183    6 0.072860874    3    0
## James     19.6892059 -3.9509075    4 0.159943750    3    0
## Jayden     6.3825643  2.0615713    7 0.021451617    2    1
## Logan     25.1085777  2.1855631    6 0.542308775    1    0
## Alexander 17.5249353 -6.7842310    8 0.047498953    1    0
## Caleb      7.5594596  1.1825783    2 0.142631623    3    0
## Ryan      11.7416278  5.0718581    7 0.017098239    0    0
## Luke      17.3704998 -8.3840656    7 0.090446829    0    0
## Daniel    18.4040739 -7.2609475    3 0.111720339    4    0
## Jack      11.7872360  2.0194711    4 0.301871504    1    1
## William    9.9815725 -1.4982323    2 0.006453232    3    0
## Owen      20.2091650  6.4915914    5 0.011335256    2    0
## Gabriel   20.9984813 -1.4287210    6 0.035902912    2    1
## Matthew   10.6431590 -9.9941786    2 0.027212663    2    0
## Connor    23.5985333 -1.2002555   10 0.200970573    2    0
## Jayce     17.3112198 -4.6498597    4 0.024323893    0    0
## Isaac     25.1913497 -6.4539217    8 0.187812729    3    1
## Sebastian 19.2601796  5.9818775    4 0.128995957    6    1
## Henry     24.0906311 -0.2932333    4 0.028278859    2    0
## Muhammad  10.3436572 -8.9193682    2 0.092533974    0    1
## Cameron   20.3262996 -4.9513969    3 0.001830488    0    0
## Wyatt     21.5982794  7.5528204    7 0.005887059    2    0
## Dylan      8.7855847  7.1488876    7 0.212643774    0    0
## Nathan     9.7185833  5.3018862    7 0.054421710    1    0
## Nicholas  12.5698115  6.2560763    8 0.084506920    3    0
## Julian    16.7086358  7.0619437    7 0.018497346    2    1
## Eli        9.4813328  6.1563604    4 0.246621867    3    1
## Levi      15.0263016 -2.5991395    4 0.011645380    4    1
## Isaiah    13.9668847  4.1343714    4 0.024058102    0    0
## Landon    27.7428360 -6.7516982    4 0.080731379    2    0
## David     24.1240825 -1.5402520    4 0.159107455    2    0
## Christian 26.3120336  3.9995236    1 0.044213118    4    0
## Andrew     5.8743114 -3.3497811    4 0.058814780    2    1
## Brayden   19.2861893  2.1287030    6 0.056114437    2    0
## John      21.1858926  0.7651000    2 0.021521685    1    0
## Lincoln   15.2239088  3.7545871    7 0.104387754    1    0
```

- Create a data frame of the same data as above except make the binary variable a factor "DOMESTIC" vs "FOREIGN" for 0 and 1 respectively. Print out the top few rows to check this worked correctly.

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

```
##                  X1         X2 X3          X4 X5       X6
## Sophia    17.8460639 -9.5060941  5 0.038361642  2 DOMESTIC
## Emma      28.4915007 -8.9494042  6 0.267481023  3 DOMESTIC
## Olivia    16.5854290 -6.7571072  4 0.320759563  2  FOREIGN
## Ava       13.6426384  3.2624797 10 0.056956998  2 DOMESTIC
## Mia       11.1561830 -1.0958796  7 0.033177328  1  FOREIGN
## Isabella   9.2997628 -5.2515940  6 0.149927116  1 DOMESTIC
## Riley     22.9104565  6.1770309  8 0.164080251  6  FOREIGN
## Aria      11.1231149  9.3543350 11 0.185602755  2 DOMESTIC
## Zoe       11.9161521  3.2380188  4 0.183038683  1 DOMESTIC
## Charlotte  7.7076312  1.2894057  9 0.126249892  2 DOMESTIC
## Lily      17.2432819  6.1564785  2 0.008464963  2 DOMESTIC
## Layla     15.0086994 -1.2397703  4 0.452499452  2 DOMESTIC
## Amelia     3.3679293 -1.8651010  7 0.108566045  4 DOMESTIC
## Emily     16.1034771 -8.1924464  5 0.298669357  5 DOMESTIC
## Madelyn   18.3381568  7.2429860  4 0.106924399  3 DOMESTIC
## Aubrey    18.8155499  3.5943590  6 0.382279928  4 DOMESTIC
## Adalyn    24.7798129  6.7852624  9 0.057037102  2 DOMESTIC
## Madison   16.5070152 -2.4699173  4 0.003585211  4  FOREIGN
## Chloe     10.4706402  8.8440410  4 0.165242959  0  FOREIGN
## Harper    26.7656433 -8.4924086  4 0.007587857  3 DOMESTIC
## Abigail   11.8064633 -1.2552792 11 0.085680424  3 DOMESTIC
## Aaliyah   11.5997922  2.6650190  3 0.134880841  3  FOREIGN
## Avery     17.1584074 -4.3875140  4 0.104212531  2  FOREIGN
## Evelyn    21.2929363  3.6682151  6 0.167579909  3 DOMESTIC
## Kaylee    18.2358792 -6.0737976  3 0.126087118  2 DOMESTIC
## Ella      15.0437353 -2.0431593  7 0.108505565  1 DOMESTIC
## Ellie     22.3032292 -3.5904466  4 0.216496446  0 DOMESTIC
## Scarlett   7.4616564  8.5137227  9 0.187972426  1  FOREIGN
## Arianna   15.9293528 -9.9830577  4 0.006363437  3 DOMESTIC
## Hailey    19.5855909  9.2276552  8 0.102149615  1 DOMESTIC
## Nora      20.2532767  4.9183549  9 0.059694295  3  FOREIGN
## Addison   33.3456553  1.5298770  6 0.020926553  3 DOMESTIC
## Brooklyn  12.2121592 -7.6243079  2 0.105940416  3 DOMESTIC
## Hannah     0.7551125 -1.5447252  7 0.144820232  3 DOMESTIC
## Mila      14.9503566 -4.3306047  5 0.250470975  2 DOMESTIC
## Leah       9.0576792  9.7354602  8 0.007902291  4 DOMESTIC
## Elizabeth 14.9684417  2.7669378  8 0.082905491  4 DOMESTIC
## Sarah     13.1405139 -4.0523690  4 0.025734279  2  FOREIGN
## Eliana    12.3896418  1.4772260  5 0.141967242  2 DOMESTIC
## Mackenzie 20.3705252  5.1017751  5 0.080281952  3 DOMESTIC
## Peyton     6.5280722  5.2138714  5 0.003543715  2 DOMESTIC
## Maria      1.7394001 -7.3650968  4 0.012500477  6 DOMESTIC
## Grace      7.1753630  1.2419093  6 0.006903284  0 DOMESTIC
## Adeline   19.4658398 -5.1258283  6 0.090066574  2 DOMESTIC
## Elena     15.4285953  9.4307998  8 0.475183383  3 DOMESTIC
## Anna       7.3502659 -1.7272058  4 0.032556350  3 DOMESTIC
## Victoria  11.0452077 -9.1667514  7 0.114661936  0 DOMESTIC
## Camilla   21.3039494 -8.7682942  5 0.568942244  3 DOMESTIC
## Lillian   21.8340344  9.3304956  6 0.001679693  3 DOMESTIC
## Natalie   10.6438132  3.6276964  2 0.005753041  4 DOMESTIC
## Jackson   27.8076653 -0.1086385  2 0.127087859  2 DOMESTIC
## Aiden     24.5353613 -8.8936180  3 0.207745144  3  FOREIGN
## Lucas     12.2851798  8.9568441 13 0.222767339  0  FOREIGN
```

```
## Liam      19.6230118  1.5540945 12 0.303399984  3 DOMESTIC
## Noah      25.8673359  5.7168356  9 0.281435189  0 DOMESTIC
## Ethan     19.6126825  4.7062299  4 0.052532391  2  FOREIGN
## Mason     15.4016300 -6.4346912  8 0.065530862  5 DOMESTIC
## Caden     13.7905921  7.9200834  6 0.268540250  1 DOMESTIC
## Oliver     7.8451608 -2.7757978  8 0.026458621  4 DOMESTIC
## Elijah    17.1831069  9.6767045  5 0.020582555  3  FOREIGN
## Grayson   18.4184724 -6.3467487  6 0.037672199  3 DOMESTIC
## Jacob     26.1388981  5.0119568  7 0.003480217  1 DOMESTIC
## Michael   26.2660677  6.0427397  8 0.042675469  0 DOMESTIC
## Benjamin  22.0588916  7.7028747  5 0.230381360  5 DOMESTIC
## Carter    23.0800879 -9.4443183  6 0.072860874  3 DOMESTIC
## James     19.6892059 -3.9509075  4 0.159943750  3 DOMESTIC
## Jayden     6.3825643  2.0615713  7 0.021451617  2  FOREIGN
## Logan     25.1085777  2.1855631  6 0.542308775  1 DOMESTIC
## Alexander 17.5249353 -6.7842310  8 0.047498953  1 DOMESTIC
## Caleb      7.5594596  1.1825783  2 0.142631623  3 DOMESTIC
## Ryan      11.7416278  5.0718581  7 0.017098239  0 DOMESTIC
## Luke      17.3704998 -8.3840656  7 0.090446829  0 DOMESTIC
## Daniel    18.4040739 -7.2609475  3 0.111720339  4 DOMESTIC
## Jack      11.7872360  2.0194711  4 0.301871504  1  FOREIGN
## William    9.9815725 -1.4982323  2 0.006453232  3 DOMESTIC
## Owen      20.2091650  6.4915914  5 0.011335256  2 DOMESTIC
## Gabriel   20.9984813 -1.4287210  6 0.035902912  2  FOREIGN
## Matthew   10.6431590 -9.9941786  2 0.027212663  2 DOMESTIC
## Connor    23.5985333 -1.2002555 10 0.200970573  2 DOMESTIC
## Jayce     17.3112198 -4.6498597  4 0.024323893  0 DOMESTIC
## Isaac     25.1913497 -6.4539217  8 0.187812729  3  FOREIGN
## Sebastian 19.2601796  5.9818775  4 0.128995957  6  FOREIGN
## Henry     24.0906311 -0.2932333  4 0.028278859  2 DOMESTIC
## Muhammad  10.3436572 -8.9193682  2 0.092533974  0  FOREIGN
## Cameron   20.3262996 -4.9513969  3 0.001830488  0 DOMESTIC
## Wyatt     21.5982794  7.5528204  7 0.005887059  2 DOMESTIC
## Dylan      8.7855847  7.1488876  7 0.212643774  0 DOMESTIC
## Nathan     9.7185833  5.3018862  7 0.054421710  1 DOMESTIC
## Nicholas  12.5698115  6.2560763  8 0.084506920  3 DOMESTIC
## Julian    16.7086358  7.0619437  7 0.018497346  2  FOREIGN
## Eli        9.4813328  6.1563604  4 0.246621867  3  FOREIGN
## Levi      15.0263016 -2.5991395  4 0.011645380  4  FOREIGN
## Isaiah    13.9668847  4.1343714  4 0.024058102  0 DOMESTIC
## Landon    27.7428360 -6.7516982  4 0.080731379  2 DOMESTIC
## David     24.1240825 -1.5402520  4 0.159107455  2 DOMESTIC
## Christian 26.3120336  3.9995236  1 0.044213118  4 DOMESTIC
## Andrew     5.8743114 -3.3497811  4 0.058814780  2  FOREIGN
## Brayden   19.2861893  2.1287030  6 0.056114437  2 DOMESTIC
## John      21.1858926  0.7651000  2 0.021521685  1 DOMESTIC
## Lincoln   15.2239088  3.7545871  7 0.104387754  1 DOMESTIC
```

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

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

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

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

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

Print out a summary of the whole dataframe.

```
summary(df)
```

```
##        X1                X2                X3              X4
##  Min.   : 0.7551   Min.   :-9.9942   Min.   : 1.00   Min.   :0.00168
##  1st Qu.:11.4889   1st Qu.:-4.7252   1st Qu.: 4.00   1st Qu.:0.02628
##  Median :16.6470   Median : 1.2122   Median : 5.50   Median :0.08787
##  Mean   :16.3749   Mean   : 0.2606   Mean   : 5.66   Mean   :0.11916
##  3rd Qu.:21.0453   3rd Qu.: 5.2359   3rd Qu.: 7.00   3rd Qu.:0.16583
##  Max.   :33.3457   Max.   : 9.7355   Max.   :13.00   Max.   :0.56894
##        X5              X6
##  Min.   :0.00   DOMESTIC:76
##  1st Qu.:1.00   FOREIGN :24
##  Median :2.00
##  Mean   :2.26
##  3rd Qu.:3.00
##  Max.   :6.00
```

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

```
n = 50
X = matrix(data = sample(c(rep(0, n^2 *.5),
                           rep(1, n^2 * .25),
                           rep(2, n^2 * .25))),
           nrow = n,
           ncol = n)
table(X)
```

```
## X
##    0    1    2
## 1250  625  625
```

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

```
for (i in 1 : n) {
  for(j in 1 : n){
    if(runif(1) < 0.3){ # runif gives anything between 0 - 1
      X[i, j] = NA
    }
  }
}
sum(is.na(X) / n^2)
```

```
## [1] 0.2956
```

```
X
```

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

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

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

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

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

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

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

```
sums = rowSums(X,na.rm = TRUE)
sums
```

```
##  [1] 26 20 25 19 24 29 28 25 34 28 25 30 25 29 32 21 27 18 24 32 31 23 20 29 26
## [26] 24 30 33 34 28 27 40 25 34 17 27 28 25 21 24 24 23 25 24 37 25 22 27 24 29
```

```
sort(sums)
```

```
##  [1] 17 18 19 20 20 21 21 22 23 23 24 24 24 24 24 24 24 25 25 25 25 25 25 25 25
## [26] 26 26 27 27 27 27 28 28 28 28 29 29 29 29 30 30 31 32 32 33 34 34 34 37 40
```

```
order(sums, decreasing = TRUE)
```

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

```
X_sorted = order(sums, decreasing = TRUE)
X_sorted
```

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

- 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.

```
std_apply_row = apply(X, 1, sd, na.rm = TRUE)
std_apply_row
```

```
##  [1] 0.8601075 0.7778445 0.7914776 0.8235612 0.8667529 0.8219673 0.8280296
##  [8] 0.9145143 0.8337397 0.8677218 0.7503501 0.8783101 0.8334409 0.8560741
## [15] 0.9158109 0.7006621 0.8461141 0.7424692 0.8238196 0.8178677 0.8992943
## [22] 0.7807787 0.7580980 0.8406761 0.8572330 0.7581490 0.8451543 0.8698761
## [29] 0.8941091 0.8693637 0.7997975 0.8988692 0.8218253 0.8337397 0.7071068
## [36] 0.8409179 0.9035482 0.8183335 0.8533253 0.8714117 0.8280787 0.8610339
## [43] 0.7503501 0.8012774 0.8883145 0.7429380 0.7359801 0.8062258 0.7611244
## [50] 0.8137537
```

```
std_apply_col = apply(X, 2, sd, na.rm = TRUE)
std_apply_col
```

```
##  [1] 0.7463518 0.8550744 0.7975517 0.8006408 0.8451543 0.9216628 0.8520859
##  [8] 0.8590129 0.8819171 0.8005155 0.8243603 0.8637067 0.8325393 0.8023076
## [15] 0.8300291 0.7603145 0.7862913 0.7620008 0.7390740 0.7622867 0.7778445
## [22] 0.9161115 0.9112246 0.8520859 0.8075276 0.6720215 0.8700899 0.8621611
## [29] 0.8451543 0.8300749 0.8206182 0.8637067 0.8742344 0.8035084 0.7100716
## [36] 0.8979456 0.8044546 0.8835413 0.7997975 0.8193951 0.8979456 0.8495145
## [43] 0.8125775 0.8517407 0.8170422 0.8243603 0.7723284 0.7663560 0.9055699
## [50] 0.8213940
```

- 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.

```
count_entires_not_zero = apply(X > 0, 2, sum, na.rm = TRUE)
count_entires_not_zero
```

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

```
count_entires_not_zero_again = apply(X == 2 | X == 1, 2, sum, na.rm = TRUE)
count_entires_not_zero_again
```

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

- 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`.

```
col_list = split(X,col(X))
col_list
```

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

```
## [26]  1  2  0  1  2 NA  2 NA  0 NA  1  0  2  0  1  0 NA  1  0  1  1 NA  0  0 NA
##
## $`3`
##  [1]  0 NA NA  0 NA NA NA  0  2  1 NA  0  0  1  2 NA NA  0  1  0  2  0  0  2 NA
## [26] NA  0 NA  0  0  1  1  0  2 NA  0 NA  0 NA  1  1 NA  0 NA  2  0 NA  0 NA  0
##
## $`4`
##  [1] NA NA NA NA NA  2  0 NA  0  1  1  2  1 NA  0 NA  0  0  0  2  0  0 NA  0 NA
## [26]  1  2  0  0  1  0  2  0 NA NA  0 NA NA  0 NA NA NA NA NA NA  0 NA NA NA  0
##
## $`5`
##  [1] NA  0  0  0  0  0  1  2  1  0 NA NA NA  2  0 NA  1 NA NA  2  0  1 NA  2  0
## [26] NA  2 NA NA  0  1  2 NA  2  0  0  2  1  0 NA  0  2 NA  0  1  0  1  2  1  1
##
## $`6`
##  [1]  0  0  2  0  0 NA  0  2 NA NA NA  2 NA  2  0  0 NA  0 NA  2  0 NA NA NA  0
## [26] NA NA  0  2  2  0 NA  2  1 NA NA NA  0  2  0  2 NA NA  2  1 NA  1  0  1  1
##
## $`7`
##  [1]  0  0  2  0  0  0  2  0  1  2  1  2  2 NA NA NA NA  0 NA  0  0 NA  0  0  0
## [26]  0  0 NA  1  1 NA NA NA  2  0  1 NA  0 NA  2  1  0  1 NA  2  1  2  0 NA NA
##
## $`8`
##  [1] NA NA NA  2  0  1  1  1  0  0  0  2 NA NA NA NA  1 NA  0  0 NA  0 NA  0  2
## [26]  1 NA  2  0  0  0  2  0  2  0 NA  0  1 NA NA  2  2 NA  0  0 NA  0 NA NA  0
##
## $`9`
##  [1] NA  0 NA NA  2  2  2  0  2  0  1 NA  1  0 NA NA  0  0  0  1  2  0  2  0  0
## [26]  2 NA  2 NA  0  0 NA  0  1 NA  2 NA NA  0  0  0  1 NA  0  0  0  1  2 NA  0
##
## $`10`
##  [1] NA  1 NA  0  0 NA  2  0  0  0  1 NA  2 NA  1  0  2 NA  0  1  0  0  1 NA  0
## [26]  1  0  1 NA  2 NA  2  1  2  0  0  0 NA  2  2  1 NA  2 NA  1  1  1  0  1  0
##
## $`11`
##  [1]  2  0 NA NA  1  0  1  0  2  0 NA NA NA  2 NA  0 NA NA  1  0  0  0  1  1  2
## [26] NA NA  0 NA  2  2 NA  0  1 NA  1  0  0  0 NA NA NA  0 NA  2  0  1 NA NA  0
##
## $`12`
##  [1]  2 NA  0 NA  0  0  1 NA  1 NA NA  0  0 NA  0  0 NA NA NA  2  0  2 NA NA  2
## [26]  2  1  0  1  1  0  0  2  0  1  0  2  2  0 NA NA NA  0  0 NA  2  1  0 NA  0
##
## $`13`
##  [1]  2 NA NA  0 NA  1  0  2  2 NA NA NA  0  0  0  1 NA  0  1  2 NA NA NA  2 NA
## [26] NA NA NA NA  0  0  2  0  2  1  0 NA  1  0  0 NA  1 NA NA NA NA  0  1 NA  1
##
## $`14`
##  [1]  0  0 NA  2 NA NA  0 NA  1  0  1  1  0 NA  1 NA  0 NA  0  1  2  0  0  0  1
## [26]  1 NA  2 NA  2  0  0 NA  1  0  0 NA  0  0 NA  0  2  1 NA  2  0 NA  0  2 NA
##
## $`15`
##  [1] NA  2  2 NA  1  0  1  2 NA NA  0  2  1  2 NA  2  1  0  0  2 NA  1 NA  0 NA
## [26]  1  0  2  2  0  1  1 NA  1 NA NA NA NA  0  0 NA NA  1  0  0  1  2  0 NA  0
##
```

```
## $`16`
##  [1]  2  2  0 NA NA  0 NA  0  1 NA  0 NA NA  1  0  1  1 NA  0  0  1  0 NA  0  1
## [26] NA  2  1  0  2  1  2  0  1  0  0 NA  0  0 NA  2  1  1 NA NA  1  1  0  1  2
##
## $`17`
##  [1]  0 NA NA NA  2  1  1 NA NA  0  0  0 NA  1 NA  1  1  0  0  1  0  0  0  2 NA
## [26]  0 NA NA  2  0  0 NA  0  0  2  1  1  2 NA  1  0  0  0  1 NA  0  2  1 NA  2
##
## $`18`
##  [1]  0  0  1  1  0 NA NA  2  0  2  0  1  2  1  2  1 NA NA  1 NA  2  1 NA NA  0
## [26] NA  0 NA  2 NA  1 NA  1  0 NA  2  2 NA  1  1  1  1 NA NA  2 NA NA NA  0  1
##
## $`19`
##  [1] NA  1 NA NA  0  0  2 NA  1  2  1 NA NA  1  0  0 NA NA NA  0  2  0  0  2  2
## [26] NA  1  0  0  0  0  1  0  1  1 NA  0  1  1  0  0 NA  1  1  0  2  0  0  1  1
##
## $`20`
##  [1] NA NA NA  0  1  1  0  1  0  0 NA  0  0 NA  2 NA  2  1 NA  0  0  1 NA  2  0
## [26]  0  1  1  1 NA  0  0 NA  0  0  2  0 NA  0  0 NA  0 NA  0  2  0  0  2  1  1
##
## $`21`
##  [1] NA  1  2  1  0 NA  1 NA NA  2  1  0  0  0  0 NA  0  0  0 NA  0  1 NA  2 NA
## [26]  0  1 NA  2 NA NA NA  0  0  0  0  1 NA  0  0  1  0  0  0  2  2 NA  0 NA  0
##
## $`22`
##  [1] NA  2 NA  1  0  1  0  2  1  0  0 NA  0  1 NA  0  0  0  2  1 NA  0  0  2 NA
## [26]  2  0 NA  0 NA  2  2  2 NA  2  2  0  0 NA  0  0  0  2  0  2 NA  1 NA  0  2
##
## $`23`
##  [1] NA  1  0  0  0  1  2  2 NA  2  2 NA NA  0  2  0  2  2  2  2  1 NA NA NA  0
## [26]  0  1 NA NA NA  0  0  1 NA  2  2  2  0  2  2  0  0  0 NA  0 NA  0  1  0  1
##
## $`24`
##  [1] NA NA  0  0 NA NA  0  0  1  2  2  0  0  2  0 NA NA  1  1  0  0  0  0  1 NA
## [26] NA  2  0  1 NA  1  2  0  0  0  0  1 NA  2 NA  2  0  1  2 NA NA NA  0  2 NA
##
## $`25`
##  [1]  0  2 NA  1 NA  1 NA NA NA  1 NA  0 NA  0  2  1  2 NA  2 NA NA  2  1  0 NA
## [26]  0  0 NA  0  2  0  0  0  0  1  0  0  1  0  2 NA  2  1  1  0 NA  1  1 NA  0
##
## $`26`
##  [1] NA  0  1  0 NA  0  1  0  0  2 NA  0  0  0 NA  0 NA  0 NA NA NA NA  1  1  2
## [26]  0 NA  0 NA  0  0  0  0  2  1  0 NA  1 NA NA  1 NA NA  1 NA  1 NA  1  0  0
##
## $`27`
##  [1]  0  0 NA  0  2  2  2  0 NA  0  0 NA NA  0  2  0  0 NA NA  2 NA  0 NA  1  0
## [26]  1  0 NA NA  2 NA  0  1  2 NA  0  0  1 NA NA NA NA  2  1  0  1 NA  1 NA  2
##
## $`28`
##  [1] NA  0  0  2  0  0  1  0  1 NA  2 NA  1  2 NA  1  2 NA  1  0  2  0  0  0  2
## [26]  2  2  2  0  1  2  2  1  0  0  0  0  0 NA  1  0  0  1 NA  0 NA  0  2  1  0
##
## $`29`
##  [1]  1 NA  0  2  0 NA  1 NA  1 NA  2  0 NA  0 NA NA  2  1  2 NA  1  2  1  0 NA
```

```
## [26]  0  0 NA  0  2  2  0  1  1  0 NA  2 NA  0 NA NA  0  2  2  0  0  0  1  1  0
##
## $`30`
##  [1]  0  0  1 NA  2  2  2  0  0  1  0  0 NA NA NA NA NA  1  0  1 NA  0  1  0  0
## [26]  0 NA  2  0  1  2  1  0  0  0  2  2  0  0  2  0  0  1  2 NA  0  1  1  1  2
##
## $`31`
##  [1]  2  1  2  1  2  2 NA  0  2  0  0  2  2 NA  2  0  2  2  0 NA NA  1  0  1  1
## [26]  0  1  0  1 NA  0  1 NA  1  0  2  1  0  0  0  0 NA  1  1 NA  0 NA NA NA  1
##
## $`32`
##  [1]  0 NA NA  2  2  1 NA  0  0  1  0  0  2 NA  0  0  0  0  0 NA NA  0 NA  0  1
## [26]  1 NA  2  0  0 NA  2 NA NA  1  2  0  2 NA  0  1  1  2  0  2  0 NA NA NA NA
##
## $`33`
##  [1] NA NA NA  2  0  0  0  2  0 NA  2  1  1 NA  2  2 NA  1  0  0  1  2  1  0  2
## [26]  1  1  1  2  0  0  0  2  1 NA  2  2 NA  2  0 NA NA  0  0 NA  0 NA NA NA  0
##
## $`34`
##  [1] NA  0  1 NA  1 NA NA  1  0  0  1  0 NA  1  2  1  2  0  0  1  0  0  1  0  1
## [26] NA NA  2  0  2  2  2  2  0  1  1  0  2  1  0  0  2  0  0  0  0  0  1  2  1
##
## $`35`
##  [1]  2  0  1  0 NA  1  0  0  1 NA  1  2  1 NA NA  1  0  0 NA  1  1  2 NA  0  0
## [26] NA  1  1  0 NA  1  0 NA  0  1 NA  2  0  0 NA  0 NA NA  1  1  1 NA  2  0 NA
##
## $`36`
##  [1]  1 NA  1  0 NA NA  0  0  0  1  1  0  1  0  2 NA  0  0  2 NA  1  2  0  2  0
## [26]  2  2 NA  2  1  0  0 NA NA  0 NA NA  0 NA  2  0  2  0  2  0  2 NA NA  0  2
##
## $`37`
##  [1]  1  0  0  0  0  0  0  2  0 NA NA  2  1  1 NA  1 NA  0  2  2 NA  2 NA  0  0
## [26]  0 NA  1  0 NA  1  1  1  0 NA  0  2  2 NA  0  0  1  2  1  0  0 NA NA  1 NA
##
## $`38`
##  [1]  1  0  1 NA  0 NA  0  2  2 NA  1  2 NA NA  2  0  0  0  2  1 NA  0 NA  0 NA
## [26] NA NA NA NA  0  0  2  2  1 NA NA  0  0 NA  2 NA  2  0 NA  0  1 NA  0 NA  0
##
## $`39`
##  [1]  0  0 NA  0  2 NA NA  0  0  1  2  2  0  1 NA  1  1  1  1 NA  2  1  0  1 NA
## [26]  0  0  0  2  0  1  2  1  0 NA  1  2  2  0 NA  0  0  0  0  0 NA NA NA  0  0
##
## $`40`
##  [1]  1  0  0  0 NA NA  0  0  2  0  0  1 NA  2 NA  0  0  0  0  1  2 NA NA  0  2
## [26]  1  0  0  0 NA NA  1  2 NA  0  1  0  2 NA  0  0  1 NA  1  0  2  0  0  2 NA
##
## $`41`
##  [1] NA  2 NA  2  0  2  2  0  1  2  0 NA  0  2  1  2  1  0  2 NA NA NA  2  0  2
## [26]  0  2  2  1 NA  2  0  2 NA  1 NA  2  1  2  0  2 NA NA  1  2  0  0  0 NA NA
##
## $`42`
##  [1]  2 NA NA NA NA  0  0 NA  2  0 NA  1  0 NA  2  0  1  0 NA NA  0  0 NA NA NA
## [26]  2 NA  0  0 NA  0 NA NA NA  2  1 NA  0  0  2  1 NA  1 NA  1  0  0 NA  2  0
##
```

```
## $`43`
##  [1]  0  2  2  0  1  1  0 NA NA  1 NA  1 NA NA  0 NA  0  2 NA NA  0  0  1  1  1
## [26]  0  2  0 NA NA  1  0 NA  2 NA NA  0 NA NA  0 NA  0  0 NA  2  1  2 NA  0  0
##
## $`44`
##  [1]  1  1  0  0 NA  0 NA  0  2  0  1  0 NA  0  1  1  1  0  0 NA  2  0  0  0 NA
## [26]  0  0  2  2 NA NA  0 NA NA  0  0 NA  0  0  2 NA NA  0  0  0  2  0  2  2  2
##
## $`45`
##  [1]  1 NA  1 NA  0  0 NA NA  2 NA  0  0  1  2  0 NA NA NA  0  0  2 NA  0 NA  0
## [26]  0  1  2  1  0 NA  2 NA  0  0  0  0  0 NA NA NA  2  0 NA  0  1  0  2  0  1
##
## $`46`
##  [1]  0 NA NA NA NA NA NA  2  0 NA  0 NA NA NA NA NA NA  0  0  0  0  1  0 NA  0
## [26]  0  2  1  2  1  1 NA  0  2  0 NA  0  0  1 NA  2  0  1  2  1  0  1 NA NA  2
##
## $`47`
##  [1] NA  0  1  0  2  1  0  0  0  0  0  0  2 NA  0  1 NA  2  0  1 NA  0 NA  0  1
## [26] NA  0  1  2  0  2 NA  1  1 NA  1  1  1  2 NA  0 NA  0  0  2  0  0  0 NA  1
##
## $`48`
##  [1] NA  1  0  0  0  2  0  0  0  0 NA NA  0  0 NA  1 NA NA NA  2  2  0  1  1  0
## [26] NA  0  2 NA  1 NA  1  0  0  0 NA  1 NA  0  0  0 NA NA  0  1  1  2  1  0  2
##
## $`49`
##  [1]  2  1  2 NA  2  2  2  0  0  2  1  1  2  0 NA NA  0  1  0 NA  0  1  2  0 NA
## [26]  1 NA  1  2 NA NA NA  0  0  0 NA NA  0  2  0  0 NA  0  2  2  0  0  2  0  0
##
## $`50`
##  [1]  2  0  1 NA  1  0  0 NA  0 NA NA  0  0 NA  2  2  0  2  0  1  1 NA  2  0 NA
## [26]  1  1 NA  2 NA  0  2  0  2  0 NA NA NA NA  1  2  0 NA NA  1 NA  1  1  1  0
```

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

```
special_list_col = lapply(col_list,
                          function(X){
                            minimum = min(X,na.rm = TRUE)
                            maximum = max(X,na.rm = TRUE)
                            pct_missing = (sum(is.na(X)) / length(X)) * 100
                            first_NA = min(which(is.na(X)))
                            c(minimum, maximum, pct_missing, first_NA)  #Needed to return all info
                          })
special_list_col
```

```
## $`1`
## [1]  0  2 32  6
##
## $`2`
## [1]  0  2 24  5
##
## $`3`
## [1]  0  2 36  2
```

```
##
## $`4`
## [1]  0  2 46  1
##
## $`5`
## [1]  0  2 28  1
##
## $`6`
## [1]  0  2 38  6
##
## $`7`
## [1]  0  2 30 14
##
## $`8`
## [1]  0  2 36  1
##
## $`9`
## [1]  0  2 28  1
##
## $`10`
## [1]  0  2 24  1
##
## $`11`
## [1]  0  2 38  3
##
## $`12`
## [1]  0  2 32  2
##
## $`13`
## [1]  0  2 44  2
##
## $`14`
## [1]  0  2 30  3
##
## $`15`
## [1]  0  2 32  1
##
## $`16`
## [1]  0  2 26  4
##
## $`17`
## [1]  0  2 28  2
##
## $`18`
## [1]  0  2 36  6
##
## $`19`
## [1]  0  2 24  1
##
## $`20`
## [1]  0  2 26  1
##
## $`21`
## [1]  0  2 30  1
```

```
##
## $`22`
## [1]  0  2 24  1
##
## $`23`
## [1]  0  2 26  1
##
## $`24`
## [1]  0  2 30  1
##
## $`25`
## [1]  0  2 30  3
##
## $`26`
## [1]  0  2 36  1
##
## $`27`
## [1]  0  2 36  3
##
## $`28`
## [1]  0  2 16  1
##
## $`29`
## [1]  0  2 28  2
##
## $`30`
## [1]  0  2 18  4
##
## $`31`
## [1]  0  2 22  7
##
## $`32`
## [1]  0  2 32  2
##
## $`33`
## [1]  0  2 28  1
##
## $`34`
## [1]  0  2 14  1
##
## $`35`
## [1]  0  2 30  5
##
## $`36`
## [1]  0  2 26  2
##
## $`37`
## [1]  0  2 26 10
##
## $`38`
## [1]  0  2 38  4
##
## $`39`
## [1]  0  2 22  3
```

```
##
## $`40`
## [1]  0  2 24  5
##
## $`41`
## [1]  0  2 26  1
##
## $`42`
## [1]  0  2 42  2
##
## $`43`
## [1]  0  2 36  8
##
## $`44`
## [1]  0  2 24  5
##
## $`45`
## [1]  0  2 32  2
##
## $`46`
## [1]  0  2 38  2
##
## $`47`
## [1]  0  2 22  1
##
## $`48`
## [1]  0  2 28  1
##
## $`49`
## [1]  0  2 24  4
##
## $`50`
## [1]  0  2 32  4
```

- Create a vector `v` consisting of a sample of 1,000 iid normal realizations with mean -10 and variance 100.

```
v = rnorm(1000, mean = -10, sd = sqrt(100))
```

- Create a function `my_reverse` which takes as required input a vector and returns the vector in reverse where the first entry is the last entry, etc. No function calls are allowed inside your function otherwise that would defeat the purpose of the exercise! (Yes, there is a base R function that does this called `rev`). Use `head` on `v` and `tail` on `my_reverse(v)` to verify it works.

```
my_reverse = function(x){
  n = length(x)
  for(i in 1 :ceiling(n / 2)){
    temp = x[i]
    x[i] = x[n - i + 1]
    x[n - i + 1] = temp
  }
  x
}

dumb_rev = function(x){
  n = length(x)
  rev_array = array(NA,n)
```

```
  for(i in 1:n){
    rev_array[i] = x[n - i + 1]
  }
  rev_array
}


head(v)
```

```
## [1] -11.059930  -5.896357  15.478425  -2.335823 -12.714922 -12.377216
```

```
tail(my_reverse(v))
```

```
## [1] -12.377216 -12.714922  -2.335823  15.478425  -5.896357 -11.059930
```

- Create a function `flip_matrix` which takes as required input a matrix, an argument `dim_to_rev` that returns the matrix with the rows in reverse order or the columns in reverse order depending on the `dim_to_rev` argument. Let the default be the dimension of the matrix that is greater.

```
#If you want to reverse columns pass in 2.
#If you want to reverse rows pass in 1.
flip_matrix = function(matrix,dim_to_rev){
  apply(matrix,dim_to_rev, my_reverse)
}
```

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

```
avg = mean(v)
std_error = sd(v) / sqrt(length(v))
```

- 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?

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

```
##        5%
## -26.02206
```

```
qnorm(0.05,-10,sqrt(100))
```

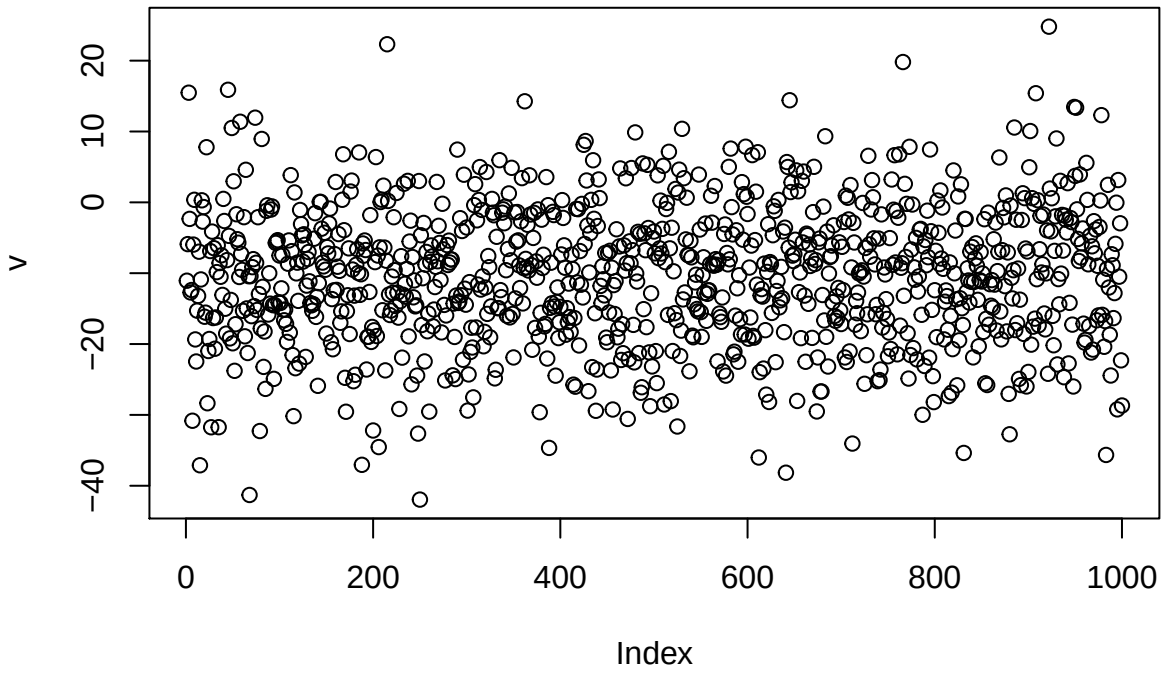
```
## [1] -26.44854
```

- 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?

```
quantile(v)
```

```
##         0%        25%        50%        75%       100%
## -41.953914 -16.721442  -9.829132  -3.846213  24.795196
```

```
plot(v)
```



- Create a list named `my_list` with keys "A", "B", ... where the entries are arrays of size 1, 2 x 2, 3 x 3 x 3, etc. Fill the array with the numbers 1, 2, 3, etc. Make 8 entries.

```
keys = c("A","B","C","D","E","F","G","H")
my_list = list()
#Iterate each key. Value be an array with nums
for(i in 1:length(keys)){
  my_list[[keys[i]]] = matrix(1:i,i,i)
}
my_list
```

```
## $A
##      [,1]
## [1,]    1
##
## $B
##      [,1] [,2]
## [1,]    1    1
## [2,]    2    2
##
## $C
##      [,1] [,2] [,3]
## [1,]    1    1    1
## [2,]    2    2    2
## [3,]    3    3    3
##
## $D
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    1    1
## [2,]    2    2    2    2
## [3,]    3    3    3    3
## [4,]    4    4    4    4
##
```

```
## $E
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    1    1    1    1
## [2,]    2    2    2    2    2
## [3,]    3    3    3    3    3
## [4,]    4    4    4    4    4
## [5,]    5    5    5    5    5
##
## $F
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    1    1    1    1    1
## [2,]    2    2    2    2    2    2
## [3,]    3    3    3    3    3    3
## [4,]    4    4    4    4    4    4
## [5,]    5    5    5    5    5    5
## [6,]    6    6    6    6    6    6
##
## $G
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
## [1,]    1    1    1    1    1    1    1
## [2,]    2    2    2    2    2    2    2
## [3,]    3    3    3    3    3    3    3
## [4,]    4    4    4    4    4    4    4
## [5,]    5    5    5    5    5    5    5
## [6,]    6    6    6    6    6    6    6
## [7,]    7    7    7    7    7    7    7
##
## $H
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]    1    1    1    1    1    1    1    1
## [2,]    2    2    2    2    2    2    2    2
## [3,]    3    3    3    3    3    3    3    3
## [4,]    4    4    4    4    4    4    4    4
## [5,]    5    5    5    5    5    5    5    5
## [6,]    6    6    6    6    6    6    6    6
## [7,]    7    7    7    7    7    7    7    7
## [8,]    8    8    8    8    8    8    8    8
```

Run the following code:

```
lapply(my_list, object.size)
```

```
## $A
## 224 bytes
##
## $B
## 232 bytes
##
## $C
## 264 bytes
##
## $D
## 280 bytes
##
## $E
```

```
## 344 bytes
##
## $F
## 360 bytes
##
## $G
## 416 bytes
##
## $H
## 472 bytes
```

Use `?object.size` to read about what these functions do. Then explain the output you see above. For the later arrays, does it make sense given the dimensions of the arrays?

#TO-DO

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

```
rm(list=ls())
```

A little about strings

- Use the `strsplit` function and `sample` to put the sentences in the string `lorem` below in random order. You will also need to manipulate the output of `strsplit` which is a list. You may need to learn basic concepts of regular expressions.

```
lorem = "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi posuere varius volutpat. Morbi

strsplit(lorem,"[.]")
```

```
## [[1]]
##  [1] "Lorem ipsum dolor sit amet, consectetur adipiscing elit"
##  [2] " Morbi posuere varius volutpat"
##  [3] " Morbi faucibus ligula id massa ultricies viverra"
##  [4] " Donec vehicula sagittis nisi non semper"
##  [5] " Donec at tempor erat"
##  [6] " Integer dapibus mi lectus, eu posuere arcu ultricies in"
##  [7] " Cras suscipit id nibh lacinia elementum"
##  [8] " Curabitur est augue, congue eget quam in, scelerisque semper magna"
##  [9] " Aenean nulla ante, iaculis sed vehicula ac, finibus vel arcu"
## [10] " Mauris at sodales augue"
```

```
output = unlist(strsplit(lorem,"[.]"))
res = sample(output)
output
```

```
##  [1] "Lorem ipsum dolor sit amet, consectetur adipiscing elit"
##  [2] " Morbi posuere varius volutpat"
##  [3] " Morbi faucibus ligula id massa ultricies viverra"
##  [4] " Donec vehicula sagittis nisi non semper"
##  [5] " Donec at tempor erat"
##  [6] " Integer dapibus mi lectus, eu posuere arcu ultricies in"
##  [7] " Cras suscipit id nibh lacinia elementum"
##  [8] " Curabitur est augue, congue eget quam in, scelerisque semper magna"
##  [9] " Aenean nulla ante, iaculis sed vehicula ac, finibus vel arcu"
## [10] " Mauris at sodales augue"
```

```
res
```

```
##  [1] "Lorem ipsum dolor sit amet, consectetur adipiscing elit"
##  [2] " Integer dapibus mi lectus, eu posuere arcu ultricies in"
##  [3] " Morbi posuere varius volutpat"
##  [4] " Morbi faucibus ligula id massa ultricies viverra"
##  [5] " Curabitur est augue, congue eget quam in, scelerisque semper magna"
##  [6] " Donec at tempor erat"
##  [7] " Donec vehicula sagittis nisi non semper"
##  [8] " Mauris at sodales augue"
##  [9] " Cras suscipit id nibh lacinia elementum"
## [10] " Aenean nulla ante, iaculis sed vehicula ac, finibus vel arcu"
```