Lab 1

Marin Azhar

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`.

```r
?which
```

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

```r
which(!y)
```

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

```r
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.

```r
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.

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

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

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

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

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

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

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

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

```r
?factor
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?

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

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

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

```r
# it come out as NA
```

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

```r
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.

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

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

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

```r
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.

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

```
## [1] 0.882
```

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

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

```
##    [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##   [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1
##   [75] 1 1 0 1 1 1 0 1 1 0 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
##  [112] 1 0 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 0 0 0 1 0 1 1
##  [149] 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 1 1 1
##  [186] 1 1 1 1 1 1 1 0 1 1 0 0 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
##  [223] 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 1 1 1 0 1 1 1 1 1 1
##  [260] 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 0 1 1 1 1
##  [297] 1 1 1 0 1 1 0 1 1 1 1 1 0 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
##  [334] 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 0 1 1 1 1 1 1 1 1 1 1
##  [371] 0 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 0 1 1 1 1
##  [408] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 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 1 1 1
##  [482] 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [519] 1 1 1 0 1 1 1 1 1 1 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1
##  [556] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1
##  [593] 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 1 1 0
##  [630] 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 0 1 1 0 1 1 1 1
##  [667] 1 1 1 1 0 1 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 1 1 1 1
##  [704] 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 1 1 1 1 1 1 1 1 0 0 1 1
##  [741] 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1
##  [778] 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 0 1 1 1 1 1 1 1 1 1 1 1
##  [815] 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1
##  [852] 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 0 1 1 1
##  [889] 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 0 1 1 1 1 1 0 1 1 0 1
##  [926] 1 1 1 1 1 1 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
##  [963] 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 0 1 1 0 1 0 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] misdimeanor felony      none        infraction  misdimeanor felony
##   [7] misdimeanor felony      felony      none        misdimeanor felony
##  [13] felony      felony      none        misdimeanor felony      none
##  [19] misdimeanor felony      infraction  felony      infraction  felony
##  [25] none        none        misdimeanor none        felony      infraction
##  [31] felony      none        misdimeanor misdimeanor felony      felony
##  [37] misdimeanor none        none        misdimeanor misdimeanor misdimeanor
##  [43] infraction  none        none        none        infraction  none
##  [49] infraction  misdimeanor misdimeanor infraction  none        none
##  [55] misdimeanor felony      infraction  misdimeanor none        felony
##  [61] misdimeanor misdimeanor felony      none        infraction  none
##  [67] none        misdimeanor felony      felony      none        none
##  [73] none        felony      infraction  felony      infraction  felony
##  [79] none        none        misdimeanor none        none        infraction
##  [85] none        misdimeanor none        none        infraction  felony
##  [91] none        felony      felony      misdimeanor misdimeanor none
##  [97] infraction  misdimeanor misdimeanor 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 =as.numeric(x_3_bin)

x_3_bin
```

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

- 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] misdimeanor felony      none        infraction  misdimeanor felony
##   [7] misdimeanor felony      felony      none        misdimeanor felony
##  [13] felony      felony      none        misdimeanor felony      none
##  [19] misdimeanor felony      infraction  felony      infraction  felony
##  [25] none        none        misdimeanor none        felony      infraction
##  [31] felony      none        misdimeanor misdimeanor felony      felony
##  [37] misdimeanor none        none        misdimeanor misdimeanor misdimeanor
##  [43] infraction  none        none        none        infraction  none
##  [49] infraction  misdimeanor misdimeanor infraction  none        none
```

```
##  [55] misdimeanor felony      infraction  misdimeanor none        felony
##  [61] misdimeanor misdimeanor felony      none        infraction  none
##  [67] none        misdimeanor felony      felony      none        none
##  [73] none        felony      infraction  felony      infraction  felony
##  [79] none        none        misdimeanor none        none        infraction
##  [85] none        misdimeanor none        none        infraction  felony
##  [91] none        felony      felony      misdimeanor misdimeanor none
##  [97] infraction  misdimeanor misdimeanor none
## Levels: none < infraction < misdimeanor < felony
```

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

```
infbin = as.numeric(x_3_ord == "infraction")
misbin = as.numeric(x_3_ord == "misdimeanor")
felbin = as.numeric(x_3_ord == "felony")
 alldata = c(infbin, misbin, felbin)
 bindata =matrix(alldata, nrow=100,ncol =3)
 colnames(bindata )= c("infraction", "misdimeanor", "felony")
 bindata
```

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

```
## [32,]          0           0       0
## [33,]          0           1       0
## [34,]          0           1       0
## [35,]          0           0       1
## [36,]          0           0       1
## [37,]          0           1       0
## [38,]          0           0       0
## [39,]          0           0       0
## [40,]          0           1       0
## [41,]          0           1       0
## [42,]          0           1       0
## [43,]          1           0       0
## [44,]          0           0       0
## [45,]          0           0       0
## [46,]          0           0       0
## [47,]          1           0       0
## [48,]          0           0       0
## [49,]          1           0       0
## [50,]          0           1       0
## [51,]          0           1       0
## [52,]          1           0       0
## [53,]          0           0       0
## [54,]          0           0       0
## [55,]          0           1       0
## [56,]          0           0       1
## [57,]          1           0       0
## [58,]          0           1       0
## [59,]          0           0       0
## [60,]          0           0       1
## [61,]          0           1       0
## [62,]          0           1       0
## [63,]          0           0       1
## [64,]          0           0       0
## [65,]          1           0       0
## [66,]          0           0       0
## [67,]          0           0       0
## [68,]          0           1       0
## [69,]          0           0       1
## [70,]          0           0       1
## [71,]          0           0       0
## [72,]          0           0       0
## [73,]          0           0       0
## [74,]          0           0       1
## [75,]          1           0       0
## [76,]          0           0       1
## [77,]          1           0       0
## [78,]          0           0       1
## [79,]          0           0       0
## [80,]          0           0       0
## [81,]          0           1       0
## [82,]          0           0       0
## [83,]          0           0       0
## [84,]          1           0       0
## [85,]          0           0       0
```

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

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

  sum of each individual row should be equal to 1(comited a crime) or zero(for no crimes)

  Verify that.

```
rowSums(bindata)
```

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

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

The column sums should look like the total number of infractions, misdimeanors, and felonys there are in a sample of 100 people

Verify that.

```
colSums(bindata)
```

```
##  infraction misdimeanor      felony
##          15          26          26
```

- 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",
```

```r
  "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"
)

M = matrix( nrow = 100, ncol = 6)

row.names(M) = fake_first_names
M[,1]= rnorm(100, mean =17, sd= sqrt(38))
M[,2]= runif(100, -10, 10)
M[,3]= rpois(100, 6)
M[,4]= rexp(100, 9)
M[,5]= rbinom(20, size =100, p= 0.12)
M[,6]= rbinom(100, size=1, p=0.24)


M
```

```
##                    [,1]            [,2] [,3]          [,4] [,5] [,6]
## Sophia    10.87084141799 -7.620454356074    6 0.3158968438518   14    0
## Emma       9.01910752113 -3.080551815219    6 0.0411177070087   11    0
## Olivia    11.44622262338  4.662724249065   13 0.0727931108947    8    0
## Ava       24.66797300249 -5.453475811519    6 0.0608780517553   11    1
## Mia        7.72774094199  8.819026080891    4 0.0879546627576    7    0
## Isabella  17.13445026348 -5.920545314439    3 0.2035329492876   11    0
## Riley      7.56561746154  3.628241410479    5 0.0763759898643   15    1
## Aria      15.26284651574 -6.700222706422    6 0.1778065348985   12    0
## Zoe       24.06230257516 -5.823445999995    3 0.0608472127157   10    1
## Charlotte  7.95448024830  0.058681345545    5 0.0812038609504   11    0
## Lily       9.09214104329 -4.929800620303    5 0.2257085999174   16    0
## Layla     21.00561338581 -7.354560359381    4 0.1047044782335    9    0
## Amelia    19.11031350595  6.954230400734    1 0.1513232782229    8    0
## Emily     19.68631630134 -6.824852116406    3 0.1922756669329    9    0
## Madelyn   15.33137731822  3.776597841643    5 0.0407521045353   10    0
## Aubrey    12.81889154824  6.589961429127    5 0.0195986428815   12    1
## Adalyn    13.14596737341 -2.414836105891    6 0.0448048533377   11    1
## Madison   13.36327977580  4.764115666039    5 0.1633635161144   10    0
## Chloe     21.15755076357  9.289039843716    6 0.0162685238756    4    0
## Harper    26.46016764523 -6.002670396119    5 0.0326955594743   13    0
## Abigail    9.48110521721 -1.349963890389    4 0.0084524795723   14    1
## Aaliyah   14.86423973612  6.437963875942   11 0.0383363361470   11    0
## Avery     18.93548647398  7.668207157403    2 0.0777795538278    8    0
## Evelyn    13.87579940832  7.415770813823    4 0.0014052161173   11    0
## Kaylee    17.02912913672  0.332856560126    6 0.2179170844776    7    0
## Ella      19.89998460088  0.083455503918    5 0.0594544003765   11    1
## Ellie     35.69396598183 -5.127922683023    8 0.0510944587489   15    0
## Scarlett  22.33953051095  9.959339150228    8 0.0908047791731   12    0
```

```
## Arianna   13.98329974549 -1.460459162481    4 0.3518035320585   10    0
## Hailey     7.91268056707 -1.526307961904    7 0.0820721346424   11    0
## Nora      19.90100248103  7.616754793562    3 0.0330518703879   16    0
## Addison   12.99710325401  6.894308910705    9 0.2243959597322    9    0
## Brooklyn  13.68086646996  0.398240662180    4 0.0991076346854    8    1
## Hannah    14.55562563593  2.217311360873    5 0.0798462334385    9    0
## Mila      13.11906331942 -5.545412516221    4 0.0899440039730   10    0
## Leah       8.54522009216  5.488408012316    7 0.1645920748419   12    0
## Elizabeth 22.57449760610  6.929300175980   10 0.0488816974167   11    0
## Sarah     21.97616712740  6.341303661466    9 0.0876847888400   10    0
## Eliana     9.36987070849  0.881463713013    8 0.2631478195663    4    0
## Mackenzie 21.21113618199 -9.860677756369    6 0.0836610563260   13    0
## Peyton     7.80729428842  5.145861022174    7 0.0505291940644   14    0
## Maria     13.25616247233  7.504017390311    6 0.0693305971929   11    1
## Grace      9.56905150487  6.447297907434    5 0.0734409693525    8    0
## Adeline    9.93866135629  3.624987038784    9 0.1895421221759   11    1
## Elena     20.06677521241  6.774886930361    7 0.3667980755182    7    0
## Anna      26.78164299802  9.052430847660    7 0.0084414238938   11    1
## Victoria  26.14308201779  7.173992232420    7 0.2909327614912   15    0
## Camilla   16.37993345482  5.069515649229    9 0.0308073195111   12    0
## Lillian   16.79185476205 -8.359192721546    3 0.0168902855791   10    0
## Natalie   13.65098980719 -0.532937231474    8 0.0585427002774   11    0
## Jackson   20.46575833753  5.321074989624    6 0.0622402181228   16    0
## Aiden     23.10060108612  4.002433624119    7 0.1335706651785    9    0
## Lucas     16.05161402103  9.459243966267    2 0.0172625056778    8    0
## Liam      29.27413915820 -1.576540041715    4 0.1578349338764    9    0
## Noah      24.09678310803 -6.893312651664    6 0.0233764042043   10    0
## Ethan      7.66450443516 -4.731625430286    7 0.0792276647102   12    1
## Mason     10.11846607273 -5.962017434649   11 0.1555491626437   11    0
## Caden     13.87187977154  4.323527077213    5 0.3518182480500   10    1
## Oliver    18.65044277317  9.860089290887    8 0.0207857384553    4    0
## Elijah    18.91858234852  4.652845626697    7 0.1094852114484   13    1
## Grayson   13.92786869115 -5.620933263563    3 0.4539050118143   14    0
## Jacob     13.56372670866 -9.131066505797    9 0.3227558798073   11    1
## Michael    7.78072736517  1.340793697163    8 0.0960077589538    8    0
## Benjamin  18.30850220536 -6.515653911047    6 0.1556799113657   11    0
## Carter    22.22022519097 -6.859387597069   10 0.0062162677808    7    0
## James      0.33323385491  3.812437984161    3 0.0950268757379   11    0
## Jayden    16.24591865017  0.599678074941    8 0.0400629003222   15    0
## Logan      7.27015969619 -2.688242504373    9 0.0774234554994   12    0
## Alexander 20.92682966676 -1.733264103532    7 0.0204314275438   10    0
## Caleb     23.76034776396 -3.956183134578    9 0.0696721594884   11    0
## Ryan      25.18421311662  9.088041279465    0 0.3335359601324   16    0
## Luke      10.15601348683 -5.593545245938    5 0.1744167601803    9    1
## Daniel    16.31270833126 -6.367807150818    5 0.0883252564771    8    0
## Jack      18.95423230802 -4.656140538864   13 0.0864841146348    9    0
## William   20.13370294199 -4.164668023586    5 0.0211014518858   10    0
## Owen      19.83702824609  4.233120372519   11 0.0535048055980   12    0
## Gabriel   21.40789811685 -4.070856231265    6 0.3417137793084   11    0
## Matthew    9.30637367080 -1.557856593281    7 0.0642879630129   10    0
## Connor    15.02971830895 -7.007462903857    2 0.0115809219682    4    0
## Jayce     21.06128846817 -0.144637399353    4 0.3121867568770   13    0
## Isaac     14.24035991810  5.557890934870    6 0.0870648394810   14    1
## Sebastian 19.50766159519 -2.031480432488    4 0.0427045901631   11    0
```

```
## Henry     16.07004916881  6.469997386448    4 0.2159702510790    8    1
## Muhammad  12.71670515862  3.401311584748    4 0.0891963266203   11    1
## Cameron   20.61835820478  4.242849601433    6 0.0749697014689    7    0
## Wyatt      9.17180600559 -9.333013338037    5 0.0161639255858   11    1
## Dylan     14.70698678311  4.070609072223    5 0.0283224839821   15    0
## Nathan    15.89140973647  5.647488473915    4 0.0205113710992   12    0
## Nicholas  22.32440656593  0.182815603912    7 0.2602150562183   10    0
## Julian    18.12855819962  1.398699372075    8 0.1277651011207   11    1
## Eli       20.88922197537  4.924859735183    6 0.1809852905478   16    0
## Levi      26.65567127924  2.952253725380    2 0.0155583824445    9    0
## Isaiah    13.66860922049 -8.088558637537    6 0.0722097356079    8    0
## Landon    18.32423158644 -9.978384855203    2 0.2240862997362    9    0
## David     25.96945596297  3.999251746573    4 0.0843861837721   10    0
## Christian 14.44793940824 -0.012656985782    7 0.0029853850914   12    0
## Andrew    22.90249187247  3.712495137006    5 0.0077320177495   11    0
## Brayden   19.38309573633 -8.291223808192    5 0.5553177977689   10    0
## John      15.26325604311  2.007378139533    9 0.0195076676706    4    0
## Lincoln   22.03513479619  4.988092295825    4 0.0404746037076   13    1
```

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

```r
df <- data.frame(M)
df$X6 <- factor(df$X6, labels =c ("DOMESTIC", "FOREIGN") , levels = c(0,1))
View(df)
```

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

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

```
##
## DOMESTIC  FOREIGN
##       78       22
```

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

```
##
## DOMESTIC  FOREIGN
##     0.78     0.22
```

Print out a summary of the whole dataframe.

```r
summary(df)
```

```
##        X1                    X2                    X3
##  Min.   : 0.33323386   Min.   :-9.97838486   Min.   : 0.0
##  1st Qu.:13.08857330   1st Qu.:-4.97933114   1st Qu.: 4.0
##  Median :16.27931349   Median : 0.74057089   Median : 6.0
##  Mean   :16.66063320   Mean   : 0.55394762   Mean   : 5.9
##  3rd Qu.:20.94652560   3rd Qu.: 5.18966451   3rd Qu.: 7.0
```

```
##  Max.   :35.69396598   Max.   : 9.95933915   Max.   :13.0
##        X4                     X5               X6
##  Min.   :0.0014052161   Min.   : 4.0   DOMESTIC:78
##  1st Qu.:0.0403716779   1st Qu.: 9.0   FOREIGN :22
##  Median :0.0795369491   Median :11.0
##  Mean   :0.1158018789   Mean   :10.6
##  3rd Qu.:0.1636706558   3rd Qu.:12.0
##  Max.   :0.5553177978   Max.   :16.0
```

- 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.

```r
#multi =rmultinom(50, size =2, prob= c(0.5,0.25,0.25))
#R = matrix( multi , nrow = 50, ncol = 50)
#R

n<- 50
x<- sample(c(rep(0,n*n/2), rep(1,n*n/4), rep(2,n*n/4))) # sample randomizes
R<- matrix(x, nrow= n, ncol =n)

head(R)
```

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

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

```r
for(i in 50*50){

  if(rbinom(1,1,p = 0.3) == 0) {R[i] <- NA}

 # R[i] == "NA"

}

head(R)
```

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

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

```r
#TO-DO
sort(rowSums(R))
```

```
##  [1] 25 28 30 31 31 32 32 32 33 33 33 33 34 34 34 34 34 34 35 35 36 36 36 37 38
## [26] 38 38 38 39 39 40 40 40 41 41 41 41 41 41 41 41 43 44 45 46 46 47 48 49
```

- We will now learn the `apply` function. This is a handy function that saves writing for loops which should be eschewed in R. Use the apply function to compute a vector whose entries are the standard deviation of each row. Use the apply function to compute a vector whose entries are the standard deviation of each column. Be careful about the NA's! This should be one line.

```r
sqrt(apply(R,2, var  ))
```

```
##  [1] 0.80913155763 0.82832508653 0.85714285714 0.70450445748 0.84006802446
##  [6] 0.82536324718 0.82338276863 0.83397254396 0.79897894023 0.87037406924
## [11] 0.90913729010 0.80533932511 0.77222181439 0.77748941853 0.80330948111
## [16] 0.77958649427 0.81215259449 0.77617586898 0.87691481447 0.80812203564
## [21] 0.83029377638 0.85332482989 0.77222181439 0.84370417487 0.84176684929
## [26] 0.79385394225 0.80025506138 0.76371808026 0.85714285714 0.91160320763
## [31] 0.88063057185 0.89647837084 0.85714285714 0.88063057185 0.80913155763
## [36] 0.82709227578 0.83909572318 0.80533932511 0.70681810725 0.86307471240
## [41] 0.86307471240 0.79179465489 0.81441101793 0.78480466828 0.85260704636
## [46] 0.78869564239 0.87341693529 0.84006802446 0.89191378736            NA
```

- 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.

```r
apply(R,2, sum)
```

```
##  [1] 36 37 40 22 39 41 33 36 44 38 45 31 33 37 37 31 28 32 46 30 31 46 33 34 42
## [26] 34 41 39 40 42 40 41 40 50 36 32 55 31 26 45 45 42 35 29 37 26 41 39 51 NA
```

- 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`.

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

```
## $`1`
##  [1] 2 2 1 0 0 0 0 2 1 0 2 0 0 1 1 2 0 0 0 1 2 2 1 1 1 0 2 0 2 0 0 0 0 1 0 1 1 1
## [39] 0 1 1 0 0 0 0 0 0 2 2 0
##
## $`2`
##  [1] 0 0 0 0 2 1 1 0 0 0 0 0 1 2 1 0 0 1 1 0 2 2 2 2 0 0 1 0 2 2 2 0 0 1 1 1 1 0
## [39] 1 0 0 0 1 0 2 0 0 0 2 2
##
## $`3`
##  [1] 2 2 0 2 0 0 0 0 0 2 1 1 2 0 0 0 2 1 2 2 2 0 0 0 0 2 1 1 1 1 1 1 0 0 2 1 2 2
## [39] 0 0 0 0 0 1 2 0 0 0 0 1
##
## $`4`
##  [1] 0 1 0 1 2 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 2 0 2 0 2 1 0 0 0 0 1 1 0 0 0
## [39] 0 0 0 1 0 2 1 0 0 0 0 2
##
## $`5`
##  [1] 1 0 0 2 2 1 1 0 0 2 0 0 0 0 1 0 0 2 2 0 1 0 1 2 0 2 1 1 1 0 0 0 2 0 2 0 1 1
## [39] 0 2 2 2 0 1 0 0 0 0 1 2
##
## $`6`
##  [1] 0 2 0 2 1 1 0 2 1 0 0 1 1 2 0 0 2 1 0 0 2 0 2 1 0 2 0 1 2 0 1 1 2 1 0 2 2 0
## [39] 1 0 0 1 2 1 0 0 0 0 0 1
##
## $`7`
```

```
##  [1] 2 1 0 1 0 0 0 2 0 0 1 2 2 0 0 1 2 0 2 1 1 0 0 1 2 2 0 0 2 0 2 0 0 0 2 0 0 0
## [39] 0 0 1 1 0 0 0 0 1 0 0 1
##
## $`8`
##  [1] 1 0 0 0 2 0 0 0 1 2 0 1 2 2 1 1 1 0 2 0 1 1 2 0 1 0 0 2 0 0 0 0 2 0 0 2 0 1
## [39] 2 2 0 0 0 0 0 1 1 0 2 0
##
## $`9`
##  [1] 1 0 2 1 0 1 1 1 1 1 0 1 0 1 0 2 1 2 2 1 1 0 0 2 1 2 2 2 2 2 0 1 0 0 2 1 0 0
## [39] 1 2 0 0 1 0 2 0 0 0 1 0
##
## $`10`
##  [1] 2 2 0 0 1 0 1 0 2 2 2 0 1 1 0 1 0 1 1 0 0 2 2 0 2 0 0 0 0 2 2 0 0 1 1 0 1 2
## [39] 2 2 0 0 0 2 0 0 0 0 0 0
##
## $`11`
##  [1] 0 1 0 2 2 1 2 2 0 0 0 0 1 2 2 2 2 1 1 0 1 0 2 1 2 2 0 2 0 0 0 0 0 2 0 1 0 0
## [39] 0 2 2 1 0 0 2 0 2 2 0 0
##
## $`12`
##  [1] 0 2 0 1 2 0 0 2 0 1 0 1 0 2 1 1 0 0 0 0 0 1 0 2 0 0 2 0 0 0 0 1 1 2 2 0 0 2
## [39] 0 0 0 0 1 0 2 0 0 1 1 0
##
## $`13`
##  [1] 1 2 1 0 2 0 2 0 0 0 1 0 1 2 1 0 1 1 0 0 1 0 2 1 1 0 0 0 0 0 2 0 1 1 0 2 1 0
## [39] 0 2 0 0 2 0 0 0 1 0 1 0
##
## $`14`
##  [1] 2 0 0 1 2 0 1 0 0 0 0 0 2 1 0 0 1 0 0 1 0 1 1 0 1 0 0 2 2 1 2 0 0 1 1 0 1 1
## [39] 1 2 2 1 0 1 0 2 0 2 0 1
##
## $`15`
##  [1] 2 1 2 0 1 2 0 0 0 1 0 2 2 1 1 2 2 1 0 0 0 0 0 1 0 1 0 0 0 1 0 1 1 1 1 2 2 0
## [39] 2 0 0 1 0 1 0 2 0 0 0 0
##
## $`16`
##  [1] 2 0 0 1 0 1 0 0 0 0 1 0 0 1 0 2 1 0 0 0 1 0 0 0 2 0 0 2 0 0 0 0 1 1 0 1 2 0
## [39] 2 1 2 0 2 2 0 0 1 1 0 1
##
## $`17`
##  [1] 1 0 0 0 2 0 0 0 0 0 0 0 0 2 0 0 0 0 1 2 0 2 1 0 2 0 0 2 2 2 0 0 1 0 2 0 2 1
## [39] 1 0 0 0 1 0 0 0 0 0 1 0
##
## $`18`
##  [1] 0 0 1 0 1 0 1 2 1 2 2 1 2 2 0 0 0 0 2 1 1 0 0 0 0 1 0 0 0 0 2 1 0 0 1 1 0 1
## [39] 1 0 0 0 2 0 0 0 1 0 2 0
##
## $`19`
##  [1] 0 1 2 0 2 0 2 2 2 0 1 0 1 2 2 0 2 0 0 2 0 1 2 2 1 2 0 1 0 0 2 2 0 0 2 1 1 1
## [39] 2 2 1 0 0 1 0 1 0 0 0 0
##
## $`20`
##  [1] 2 0 0 0 0 0 0 0 0 1 1 0 0 1 2 0 0 0 2 0 0 1 0 1 0 0 2 2 0 0 0 2 2 0 0 0 2 0
## [39] 0 1 2 0 2 0 1 1 1 1 0 0
```

```
##
## $`21`
##  [1] 0 2 2 0 0 0 0 2 1 0 0 1 0 0 1 0 0 1 2 2 0 0 2 2 0 2 1 0 0 0 1 2 1 0 0 0 0 0
## [39] 0 2 2 1 0 0 0 1 0 0 0 0
##
## $`22`
##  [1] 1 0 1 1 1 1 1 2 0 0 2 0 2 2 1 2 2 0 0 2 2 2 0 0 0 0 0 1 0 0 2 0 0 0 0 1 2 0
## [39] 1 0 2 2 0 2 1 2 1 1 2 1
##
## $`23`
##  [1] 0 0 1 0 2 1 2 1 2 0 1 1 2 0 0 0 2 0 1 0 1 1 0 0 1 0 0 0 0 2 1 2 2 0 1 0 0 1
## [39] 0 0 1 0 1 2 0 0 1 0 0 0
##
## $`24`
##  [1] 0 0 2 2 1 0 0 2 1 0 1 0 0 2 2 0 0 0 2 0 0 1 0 0 0 0 1 1 0 1 1 2 0 0 0 2 2 2
## [39] 0 0 0 0 0 0 0 1 2 2 0 1
##
## $`25`
##  [1] 0 0 1 0 0 2 0 0 1 2 0 0 2 2 0 2 1 1 1 2 2 0 0 2 0 0 0 0 2 0 2 1 1 1 0 0 0 1
## [39] 1 0 2 1 1 0 2 2 2 1 1 0
##
## $`26`
##  [1] 0 2 0 1 0 0 0 2 2 0 0 0 0 0 2 0 0 1 2 0 0 1 0 1 1 0 0 1 0 2 1 0 0 1 0 2 1 1
## [39] 2 0 0 2 1 0 1 1 1 0 2 0
##
## $`27`
##  [1] 0 0 2 0 0 2 2 1 0 2 0 1 0 0 1 0 1 1 0 1 1 1 0 1 0 0 0 1 2 2 1 0 2 1 0 2 0 0
## [39] 1 2 1 2 1 2 0 1 0 0 2 1
##
## $`28`
##  [1] 0 1 1 1 1 0 1 0 0 2 1 1 0 0 2 2 1 2 2 0 0 0 1 2 2 0 0 1 0 1 0 1 2 0 2 0 1 1
## [39] 0 0 1 0 1 1 1 0 2 0 0 1
##
## $`29`
##  [1] 0 0 2 2 0 2 0 1 0 2 0 2 0 1 2 0 2 2 0 0 0 2 1 1 0 1 0 0 2 2 1 0 0 0 2 1 0 0
## [39] 1 0 0 1 0 1 2 0 1 1 0 2
##
## $`30`
##  [1] 0 2 2 0 0 0 2 0 0 0 2 2 0 2 0 0 0 2 1 2 0 2 0 1 1 1 1 0 0 0 0 0 0 0 2 0 0 0
## [39] 2 0 1 2 2 0 1 2 1 2 2 2
##
## $`31`
##  [1] 0 2 1 1 0 2 0 0 1 0 0 1 0 0 0 2 0 2 0 2 0 0 1 2 2 2 1 0 2 0 1 2 0 0 2 0 0 0
## [39] 2 1 1 2 2 0 0 0 2 0 0 1
##
## $`32`
##  [1] 0 0 0 0 0 2 0 0 0 0 1 0 0 2 2 1 2 2 2 0 2 0 2 1 1 0 0 2 1 2 0 0 0 0 1 0 0 0
## [39] 1 1 0 2 2 2 2 2 2 0 1 0
##
## $`33`
##  [1] 0 2 0 1 0 0 1 0 0 2 1 0 2 1 2 0 1 2 2 0 0 0 0 0 0 2 2 2 0 2 2 2 0 2 0 0 1 0
## [39] 1 0 0 1 1 0 1 2 1 0 1 0
##
## $`34`
```

```
##  [1] 0 1 1 0 0 2 2 1 0 0 0 2 1 0 0 0 2 2 0 0 1 2 2 0 2 2 2 1 1 0 0 2 0 2 1 2 1 0
## [39] 1 0 1 2 2 2 2 2 2 1 0 0
##
## $`35`
##  [1] 2 2 1 0 0 1 0 1 2 0 0 0 2 1 2 0 0 0 0 0 2 0 2 0 0 1 2 0 0 1 1 0 2 0 0 1 2 0
## [39] 1 1 1 0 0 1 0 2 1 0 0 1
##
## $`36`
##  [1] 0 0 0 2 0 2 1 1 0 0 2 0 0 0 0 1 2 1 0 0 2 0 1 2 0 0 1 0 2 1 1 0 0 0 0 0 2 0
## [39] 0 0 0 1 2 0 1 0 2 0 0 2
##
## $`37`
##  [1] 2 2 2 1 0 1 0 1 2 1 1 2 2 2 1 2 0 0 0 0 1 1 0 0 2 2 1 0 2 0 2 2 2 0 2 0 2 0
## [39] 2 1 1 2 2 1 0 1 1 0 2 1
##
## $`38`
##  [1] 2 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 2 2 2 0 0 0 1 1 0 0 1 2 0 0 0 2 2 2 0 0 0
## [39] 0 0 0 1 1 0 1 0 0 0 2 2
##
## $`39`
##  [1] 0 0 2 0 0 0 1 2 0 1 1 0 1 0 0 1 1 1 1 1 0 0 1 0 2 0 0 0 0 0 0 0 0 0 1 2 0 0
## [39] 2 0 0 1 0 1 0 2 0 1 0 0
##
## $`40`
##  [1] 0 2 2 0 0 1 0 2 0 2 1 1 2 0 1 1 0 1 0 0 2 1 0 0 2 0 2 1 0 1 1 0 0 2 0 2 2 1
## [39] 1 2 2 0 2 2 0 0 0 2 1 0
##
## $`41`
##  [1] 0 0 0 2 0 0 0 2 1 2 0 1 0 1 0 2 0 2 2 0 2 1 1 0 1 1 2 0 1 0 1 1 0 1 0 2 2 1
## [39] 2 2 0 1 0 0 2 2 0 2 0 2
##
## $`42`
##  [1] 0 1 1 2 0 1 2 2 0 2 2 0 1 1 1 0 0 0 1 1 1 0 1 0 0 1 1 0 2 2 0 1 2 1 2 2 1 0
## [39] 2 0 0 0 0 0 1 0 0 1 2 1
##
## $`43`
##  [1] 2 2 0 1 1 1 0 1 0 1 0 0 2 0 0 2 2 0 0 0 1 1 2 2 0 0 0 2 2 0 0 0 0 1 0 0 0 1
## [39] 0 0 1 2 0 0 0 2 1 1 1 0
##
## $`44`
##  [1] 0 0 0 1 2 0 0 0 0 0 0 2 0 1 0 2 1 1 0 0 0 0 0 2 1 2 1 1 2 0 2 0 0 0 2 0 0 1
## [39] 0 2 1 0 0 0 0 0 0 1 0 1
##
## $`45`
##  [1] 0 2 1 1 0 0 1 2 1 2 2 2 0 2 0 0 2 0 0 0 0 0 0 0 2 0 0 0 2 2 2 1 0 2 1 0 0 1
## [39] 0 0 1 0 0 1 0 2 0 1 0 1
##
## $`46`
##  [1] 1 2 0 0 0 0 0 1 0 2 0 0 0 0 1 0 0 1 0 2 0 0 2 1 0 0 1 2 2 0 0 0 2 0 0 1 0 0
## [39] 0 0 1 0 0 0 2 2 0 0 0 0
##
## $`47`
##  [1] 0 1 0 0 1 0 2 0 0 1 0 2 1 1 0 0 0 0 0 2 1 2 0 0 2 2 2 0 2 0 2 0 0 2 2 1 2 1
## [39] 0 1 2 0 2 0 0 0 0 2 1 1
```

```
##
## $`48`
##  [1] 1 0 1 0 0 0 0 1 1 1 0 2 0 2 2 1 2 0 0 2 0 1 2 0 0 2 0 2 0 0 2 2 1 1 0 2 0 1
## [39] 0 0 0 0 1 1 2 0 0 0 1 2
##
## $`49`
##  [1] 0 2 0 0 2 2 0 1 1 0 1 0 2 0 2 0 0 2 0 1 2 0 0 1 1 2 0 1 0 1 2 0 2 0 2 2 1 2
## [39] 0 0 1 2 2 2 2 0 1 2 2 2
##
## $`50`
##  [1]  0  2  0  1  2  1  2  2  0  0  0  1  1  1  0  0  2  0  2  2  2  0  0  0  1
## [26]  2  0  1  0  0  0  0  0  1  1  0  0  0  0  0  0  0  0  1  2  2  2  0  1 NA
```

- 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.

```r
#a<-lapply(R,min)
#b<-lapply(R, max)
#lapply(R,if( is.na == TRUE) print )

#c<- matrix(c(a,b) ,50*50,2)

#c
```

- 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.

```r
#TO-DO
set.seed(10)
v = as.vector(rnorm(1000, -10, 10))
v
```

```
##    [1]  -9.8125382905817 -11.8425254206906 -23.7133054992251 -15.9916771578372
##    [5]  -7.0545487343249  -6.1020569929983 -22.0807617542949 -13.6367601747086
##    [9] -26.2667268170309 -12.5647839412399   1.0177950308713  -2.4421849197266
##   [13] -12.3823355601872  -0.1255529658661  -2.5860987161618  -9.1065273350418
##   [17] -19.5494385615238 -11.9515038466724  -0.7447873790592  -5.1702147516339
##   [21] -15.9631063672021 -31.8528683816953 -16.7486593787512 -31.1906119191017
##   [25] -22.6519802153090 -13.7366155515470 -16.8755543038792 -18.7215882671769
##   [29] -11.0176100622482 -12.5378053010246 -28.5374045447914 -10.7794606607537
##   [33]  -0.3143365947546  -8.1507404000969 -23.7994357833758 -24.3551436236037
##   [37]  -6.3791277139339 -27.5908675375971 -13.2454400957233 -16.5156298854466
##   [41]   0.8655139944051 -17.6254488003129 -18.2866253500109  -1.6552609691155
##   [45] -19.6765198675991 -10.2881533547586  -7.6747484742461 -13.0120868150418
##   [49] -16.7761458314933  -3.4477236376478 -14.0063754703174 -13.3455656507335
##   [53]   3.6795395319196  11.3776710365012  -4.9418073547097  -2.1365761576084
##   [57] -19.0221194417864  -4.6710300767167 -16.4589425354921  -7.0901251157023
##   [61] -22.3759446887722 -14.5617627511781 -18.3032265472473  -6.5988435632574
##   [65]   0.6637639568217   2.1612583807980  -2.6430934236695 -14.8120861731558
##   [69]  -4.3725523714188 -22.4631971188920  -6.1907778737432 -24.3042725279669
##   [73] -20.4844550487860 -12.1850355053459 -24.8993623673554   1.7270628121431
```

```
##  [77] -24.7982702157166 -14.3038781607722 -20.5163864204345   5.2258634405410
##  [81]  -4.0717194541392 -12.2266150901927  -2.8710572375154  -2.8339916625895
##  [85]  -5.5975813561586  -8.4116937868190  -3.4023586166805  12.2051966293556
##  [89] -21.8394507406540 -10.7395583449749 -14.1635467488652 -11.9148234375263
##  [93]  -9.3045521859260   1.5534831801052  -4.0504265304951 -24.1964510835699
##  [97] -26.0667724535804  -1.0707410043682  -8.5183204481227   2.2702839010139
## [101] -17.6180433917803  -5.8062459411009 -20.3994336463235  -2.8842603400737
## [105] -16.3321301496783  -4.3682533554985  -3.3901331416840 -26.5805085732545
## [109]   0.2816797701792   1.2795361401459 -22.8015460342218   1.2886822740957
## [113] -14.6413452716498 -13.1576020953137  -0.7570685316505  -9.2285527601422
## [117]   0.3992360511188  -2.5811379326182   2.5554485828951  -0.4908103354382
## [121] -14.8136560727329  -7.9711822203016 -10.3173974383773 -21.9558030033457
## [125]  -3.7631876315157 -19.1480448366691  -7.5124199229190 -20.6262279318038
## [129] -13.6398224719576 -22.0699485337827   4.2921278138977  -3.6656410901717
## [133] -29.9681561765642 -16.8183217309621 -14.6005547931070 -19.8306919414776
## [137]  -5.0466828711166  -2.7418249976747  -3.3270126810708  -0.4521356353341
## [141] -26.7533217929194 -22.0518539249191 -29.6325248922053   4.7075230981397
## [145]  -6.2752766144940   0.6587933403768  -4.6935013164268  -8.9801655411587
## [149]   3.3778246578648  -9.1276523150887 -13.9110420740045 -12.4986748459556
## [153]   1.5510474589615 -18.6472723983100 -18.6667834236831 -33.2101703034786
## [157]  -3.9116983108215   1.5000604820099 -21.9959767173686 -25.8000075454988
## [161]  -3.4683380635781 -15.4940848512297  -4.7894547468747 -16.9940306640950
## [165] -14.3890931479220 -16.7731929611669  -0.4085880522465 -24.6817332946240
## [169]  -8.1623610733187 -24.3514718265180 -21.3739989812347 -14.1464532687800
## [173]  -8.5606571184864   0.6202433069828 -15.7079390273540   2.7718137641215
## [177]  -7.7171067945992 -13.0881306450274  -0.4017086940442  -4.5117762518792
## [181]  -5.7448690622802  -3.5649996487183 -23.6030614350351 -11.9850610640319
## [185]  -3.8069732312462  10.6820960503683 -13.0528475419725  -7.1875438777014
## [189]  -3.0868266329410  -9.5363856185668  -8.8697063837404  -0.0466812571087
## [193] -16.8115136132165 -22.7705724668769 -24.6869774983108 -13.1347406653587
## [197] -27.0365949266331 -23.5051465613771 -21.0209367720916 -20.9954301452935
## [201]   2.1551377619565  -6.6912351489586   3.9027511927468  -1.2795301497889
## [205] -20.8081702266027  -5.0417840855346   0.5262755621631 -22.7464995015128
## [209] -11.9366672842338 -22.9508364255337  -8.5811971532135   2.6171505198662
## [213] -14.3150032806965 -28.2271258950182  -6.4745604124689 -23.4845144147378
## [217]  -2.9231168209257 -14.1089093679878 -14.4604517992225 -20.4115630166875
## [221] -13.2922471615401 -12.8282162413235  -5.6757087450465 -13.0760709506342
## [225] -10.5663630788932  -2.6648457979973  -9.0268838033482   6.3089173658816
## [229]  -4.3938930099780   3.2956476445216 -12.7882349563446 -22.6673154414092
## [233] -12.4914839404027  -9.8201158543120  -6.2292727288967  -2.0399144247256
## [237] -18.4067741709857 -32.0547175082621 -21.2805599106249 -23.4130995880245
## [241]   6.0511404288097  -2.5557627151146  -1.3791779404662  -6.0484418452380
## [245]  -4.9088130366967 -11.2255012583063  -9.0741535282172 -13.5787991433363
## [249] -13.5965522434701   0.2857072088428   0.7789259245098  -0.6821878476750
## [253] -24.6079386697760 -19.0607557624934 -16.8034783139241   0.6316603678904
## [257] -16.9241315212615 -21.3362826529090 -20.9461543379722 -20.1290361883673
## [261]  -5.8972267411411  -5.2225955807321 -33.2987954506654  -9.8374453482326
## [265]  -0.1964647685352  -1.9365173518264  -8.8039911110870 -34.3661560431758
## [269]  -9.6897752387761 -13.3574669799984 -12.6377616192376 -13.5585246088020
## [273] -14.8115479829018  -7.7726453218213  14.2991031869739   4.9629811873421
## [277] -17.1720820275909 -14.6705390018763  -3.3707654714059  13.0017689094660
## [281]  -6.7248985878341  -9.3613651838309 -21.3956093996767   1.8041020362035
## [285]  -9.5861152296685 -22.1359356338470  -9.2680422337219 -12.5732459434554
## [289]  -7.3319358905787   3.8772432571642  -8.0692036497606  -4.0768337739957
```

```
## [293] -18.2997452691578  -6.0742671835751  -6.1513238931919   0.5104466572565
## [297]   1.5579748458375 -20.3443795894203 -12.5446806580427   2.7368426608477
## [301]   5.0254463898139  -4.0959053285336 -16.3068545138345  -2.0765046270573
## [305]  -8.7461542987692  -6.7724502875197 -14.4558320945259  -2.3315606553862
## [309] -24.0350302182772 -21.7604677961141  -4.8840353081520   3.1676526377830
## [313]  13.9291294709557 -10.6776940849240 -12.0378506645272   9.9643445216969
## [317]   5.4321779434101 -22.2833762626182  16.4448799222478  -2.4298629172563
## [321]   0.5534519215293  -0.4205510469419  -1.6612094235735 -11.6927557001864
## [325] -14.0443364307844  -7.0439842629709 -12.3187450723486 -14.4807823162639
## [329]  -9.7768920342761  -9.5553270276387  -5.1927883470936  -3.6438014483043
## [333] -30.0074239307650 -16.9065550376296 -11.3064518930652 -13.4425178302364
## [337] -10.5241282396513  -9.0387607945088  -7.3359294676787  -4.4520652472126
## [341]   2.3544587854851  -7.1486661869311 -14.5879292777288  -3.7652197813763
## [345] -17.2399509164038   6.1941870348572 -16.1667341021694  -6.1545894255312
## [349]  11.1452131280045 -20.2856736628788 -18.8678796566786   2.7114600133104
## [353] -26.0508544127984   1.2227338293318  11.5843864176903  -5.7175338697736
## [357]   2.0117869045970   0.3169011615184  -3.4612574728974  10.1208177317955
## [361]   0.9462984183357  -2.7359064591801 -23.9965713658319 -22.8779516329080
## [365] -19.6841547752596 -27.3668575771840  -1.6830264192646 -17.5895310856506
## [369]  11.5507700106696 -14.2325888023502 -10.0564993667530  -5.9963827138848
## [373] -13.6567780204369 -14.2914383522530  -7.5530287320270  12.1977624873515
## [377]  -6.8573019208592 -15.6378507451632 -24.8569615831883  -5.8470140690619
## [381] -15.2806368267859 -13.5047061745097  -2.6937806270378 -15.5662416507982
## [385] -24.4976053616599 -20.5199642811248 -16.8283861878505 -16.1294117857592
## [389]   8.9078003505454 -11.4541053359420 -16.3942372058860 -14.1117604522169
## [393]  -6.5720812156793 -21.3324675462843  -9.3392969748082  -9.6200023403604
## [397]  -0.7893525245555  11.3496563845849 -15.9982693155619   2.4444645863646
## [401] -10.7945701712045   1.8175154981561  11.8614406267608  -5.9382506929464
## [405] -17.3835910458486 -29.5648670291891 -29.5004570159855 -19.4099781569260
## [409]   1.9778676862693 -16.2423939035829 -11.3286965510209 -10.1722092219131
## [413] -14.5962928217906   4.7293679052688  11.6931961261593 -40.0143130631433
## [417] -27.7198558305170 -13.6483922283578  -6.2546621931091 -22.3407792667110
## [421]  -5.2510657804091   2.3305745705665 -15.9312865679849  -2.4773756067248
## [425]  -3.8795449311322 -12.2865736527439 -17.0483934989938   2.2951604720639
## [429]   5.1851512986719   3.4509418039941 -11.2369152893043  -0.6609754644998
## [433]  -9.8050315437978  -1.6914789025703 -13.0449466799607 -10.5928127710359
## [437]   0.0739669737882 -23.7779508602292 -20.0420272696930 -10.2328030491504
## [441]   5.4674727378832 -17.5936325102132   0.0679265745540  -7.2578508365171
## [445] -10.8038236456008 -19.0355249532682  -5.2685489702804   3.8848897790595
## [449]  -9.6525390623583 -22.9258502807925   1.6396750490781 -25.2338459706839
## [453] -35.1833505361223 -17.0752923056637 -12.8743288270911 -14.3530831147691
## [457] -13.4307877074330 -10.3931088400990  -1.1303401227621 -13.5994840081589
## [461] -17.2895427063139   7.5174561927189 -10.8603859023211   1.0636301587614
## [465] -10.1917783723957 -24.9346017845903  17.0063662306933 -17.4369442224344
## [469]   0.6519498093684 -10.3208173166496 -15.3995968933919 -29.2564356755388
## [473]   0.0783137615909 -12.7923479676676 -22.4956585528460   2.4849112786506
## [477] -11.9221337656587  -0.8802376891256 -31.5826968002626   2.8961053233286
## [481] -14.2810996555383 -12.5500289297295   3.2715823880373  -1.6130987938999
## [485] -24.6500067717319 -16.1625269787270  -8.9389416059358 -28.4401675772715
## [489] -26.1642285371569  -7.7232329586370  -0.7878329806117 -13.3120343949900
## [493]   2.8723403098892  -9.4273919754899 -32.0393312516692 -15.2291034826556
## [497]   0.8078531012050 -12.4314133761565 -19.0944656225310 -18.6829950737599
## [501]  -1.3052495509230 -16.8000959781380  -8.2678546129149 -11.5943803907549
## [505]  -2.0650056735672   6.9435049142761   2.3996868635003  -9.7056325685223
```

```
## [509]   6.5721449904792   1.3122314018687 -24.0241063212584  -8.4322953888707
## [513]  -1.3249140356862 -10.3323515710174  -0.4593544367399   1.3009672517014
## [517]  -6.0630268791701  -6.2319167203290 -12.5174290704864  -4.9590510917322
## [521] -23.2980347134830 -11.1829945251477 -16.3324846660885  -3.2230865528305
## [525] -12.9788176477430  -3.7207775287270   6.7843072799152 -34.0000254062929
## [529] -23.4984288507140   2.0200789578675   2.9384271937366 -10.5959434618245
## [533] -16.8506234517232 -24.5343930439156   2.9412656120527   1.7616908935200
## [537]  -1.3595492182401 -32.9673252346068 -23.9181001193225 -23.8587674502157
## [541] -20.7304684391546 -19.4142550319634 -15.8643430972218   0.3472122005478
## [545]  -5.4103779330522 -21.9344507199800  -2.0811072832840 -27.7230491079952
## [549] -15.8772242199909 -22.8237031255411 -13.1213466098008 -16.9877530062472
## [553] -36.4384976770901  -1.7497762320080  -6.9307368678959 -10.5803945817092
## [557]  -0.7343058916559 -12.1869186057380  -0.0686650257996 -29.2333434248037
## [561] -20.5271518447267  -1.0960223807401  25.4114027762577  -2.5848925731212
## [565] -12.0747125409922  -8.2006154580829   4.2636484940947 -14.4316678948262
## [569]  -6.3281001334818  -3.0417830388934  -0.6655527138822   4.9188908255981
## [573]   1.5952299023330  -7.7320404309088 -22.9546518474018 -10.6277869169191
## [577]  -0.0671706312629 -20.0593093995471 -17.5098494759810   7.4401638235168
## [581] -21.8760009145738 -14.1697041771968 -25.6982015936441 -12.8047403525336
## [585] -15.8532696388542  -3.3968080243666 -16.3751254898457 -15.3902163321128
## [589]  -0.4847976644570 -11.0557984584728 -13.7406927276919 -11.7623258082391
## [593]  -7.2351669207907  -7.6366740733879  10.6167415943736 -22.9005998583527
## [597]  -4.2881055252591   7.8539022670878  -7.8705975690789  -1.1530785394932
## [601] -19.0197026441077 -10.5474995804949 -23.5675471109976  -6.8276942287425
## [605]   3.9571911930986 -22.1122381679601 -12.0414166079742 -25.5245132607939
## [609]  -4.8195033138507 -14.1884030323838 -27.0945321567921  14.8316267137122
## [613]  -5.0196626493359 -16.7386820519833 -18.8698139561134 -27.3137735086099
## [617]  -3.9545087863572 -18.7127507822903  -2.7465835362727 -10.3073198012791
## [621] -11.1722237431415  -9.0177706141557 -16.7685722184453 -10.4924635492814
## [625] -15.6071975993502  13.9915486524696 -10.8987210159804 -18.2164841778561
## [629] -12.8696176159712  -8.6872112602923 -20.6208395043809  -2.6641896427090
## [633]  -6.7730780152115 -13.4897977134871   2.8352566424672 -20.1448804888177
## [637] -15.1006427410707 -11.6716974615534 -13.0266778560616  10.2647047246020
## [641]  -0.9909436158995  -8.6457095954416 -12.0128895419938 -18.0683674099314
## [645]  -3.5587587624079 -13.3665912065361 -27.3170351124497  -7.3807933739103
## [649] -19.1669759810932 -15.6856447514912 -18.6582044175357  -1.7450608820195
## [653] -18.0013228420733  -0.2416991321562  17.0007553074187 -11.3538959688570
## [657] -11.0603116395947 -23.4406380514536  -5.5687817138720  -0.3888225833660
## [661] -12.4143581665569  -9.1704510889171 -13.6172475776824  -6.6613980019877
## [665]  -7.3439057902948  20.1290574660275 -13.4473013768743   0.6255745328506
## [669] -17.2968507404050  -9.4345227902359  -5.3217028807995 -13.7264525055132
## [673]  -7.8983831230669   2.8960830567623 -26.2683954986708 -17.8353650968338
## [677]  -4.8908689654305   2.1762605593289  -6.9489075231166  -7.8970636667506
## [681]  -5.3113403449412 -29.0434257226243   2.3840228464858  -6.0900110854235
## [685] -10.3119891674686 -26.1880052176578  -5.4873372186718 -16.9084336362306
## [689]   4.0544562193991 -27.0134268782504  -8.0775440272286 -17.0295676149723
## [693] -18.0855586334880 -16.3819874152539   7.3734542796118 -32.6047826314218
## [697] -26.1247209388646  -3.4625980119919  -0.9711927140699  -2.4539151216549
## [701] -12.2681076877007 -16.9596862759163  -4.9335264266620  -3.8376570639239
## [705] -26.6687927451536 -13.3625422574265  -4.9282014325343 -26.9739070150845
## [709] -12.0692722390273 -13.9859702817616 -21.8861982748089  -5.4743459586091
## [713]  -7.9241029607058 -22.1012237611841  -2.3258494571936 -15.9902769125696
## [717] -25.6097277280565 -19.9566404292894   0.9325100838442  -0.8285711399263
## [721] -17.7304935513235 -14.7583581764580 -21.3070318401985 -14.2883543166679
```

```
## [725] -18.3958364986366 -16.0098667290394 -17.7389077819077 -18.9249235117396
## [729] -10.3573184705772 -13.0417810239351 -40.1216378355869 -30.0852305776173
## [733] -10.4637140944499 -13.5953706450041 -11.9502166174765 -25.1447013546087
## [737] -18.9203304712062 -18.6270649141407  -4.4804585204538 -11.8823391331742
## [741] -13.2514674362955 -15.3693844913960 -13.7147741274935  -2.7719297634096
## [745]   2.0569188339316   1.8415312566266  -3.2880956459256   1.7550207341849
## [749]  -8.1147352727879  -5.0093640315497 -29.4761831141199  -5.7926275382354
## [753]   1.7347624977911 -24.4623155699525 -13.6675120594831 -16.0779623487786
## [757]  -6.8012862261922   0.9984767220601 -14.0603662759087  14.1074642682032
## [761]   1.5889988480835  -6.6161040844377  -5.4382265607910 -13.1404466641558
## [765]   4.0434402019996 -14.1026994908487 -10.8810251615009  -6.7512440528369
## [769]   1.8509980019251 -12.5550857344088  -1.1555207910437 -14.8856106962647
## [773] -17.2737617317756 -21.8825527291670 -15.3794241591258  -8.8064878268304
## [777]   3.1817809545599   4.4308055765009  -7.5097378835476  -1.8429767314940
## [781] -17.3460780637786  -7.4184672197781 -18.5286076882271  -7.8435124144653
## [785] -17.0921021200240  -2.8913447684106 -14.4737185876207 -13.3829830419350
## [789]  -4.8276657785930  -9.0129640567598  -2.5631438416316 -12.7855014148461
## [793]  -4.6092831938205 -10.8833946967005 -17.4575750815326 -25.6007845110139
## [797]   0.8295652542379 -33.4105600256368  -7.9369473396797  -2.3525742881898
## [801] -31.2506858104394  -4.7882133369683 -10.4642289995611 -21.4947172097093
## [805]  -3.7066169229980 -17.2647147599165 -13.8641732230971 -16.7171438988431
## [809]  -0.9961380963144  -1.1315830938367  -6.0972188810054 -19.2751149022227
## [813]  -2.5416077121940  -5.0517584335736 -10.2193146172307  -4.1197767373419
## [817]  -2.4350892599763 -14.9402627261870 -17.3331562861347 -10.0102039398779
## [821]  -1.9417430322349   1.0930938187503 -13.0967948215953 -19.0857824812747
## [825]  -7.3697387231332 -10.2091780757642 -12.1234250764778 -20.3110391977497
## [829]  -8.6048926030375  -8.3378201538353 -19.8465964905627  -9.7736592922938
## [833]  -2.1466996733904 -21.7918952425080   9.1079077756379 -15.3800149491958
## [837]  -1.1572357097661  -8.5269036329116 -20.0842736285689  -1.6181100779241
## [841] -21.6823913740479  -5.4568186708000  -7.8742713249553 -15.5308634742956
## [845]  -9.7656997482033  -3.5182110960861 -27.4279473520386  -4.0483733908813
## [849] -18.7852901298319  -3.5556242870789 -16.2909850266660 -10.9820018968384
## [853]  -3.1578230083224 -23.8468603099041   5.5494558598051 -18.1434134188457
## [857] -18.4363108589257   0.2492742587361  -8.8429662360384  -7.5850596012720
## [861] -23.9081614168685   5.2606084457092  -9.5348376327860   2.3998235965392
## [865]  -6.9142736453721  -2.1437239549793   1.1747870473614 -10.0498227234439
## [869] -11.7841821019019  -3.0377093105292  -4.7820279150645 -14.4782659024346
## [873]  -5.9459939595531 -17.3696442466941  -7.4425198558953 -12.8493350493357
## [877] -10.3112132499334 -14.8585888696012 -11.3508808415585 -24.8071872283751
## [881]   2.5636912568560  -7.9095835009043  -6.8684321040987   0.7514823866383
## [885] -12.8662517457042 -10.3197742743150 -17.4273339663881  10.8522019215140
## [889]  -1.4510958959000   0.0044599614444 -21.8411791008532 -25.4110255512378
## [893] -12.1173300076082  -8.2964743003733 -16.9077737443293   6.9772073903203
## [897] -10.7268996260587 -17.1760885628929 -14.1651443830786  11.9358676105577
## [901]  -4.5765217421848   0.2366760898180  -1.8717389128953   3.0102707741098
## [905]  -4.3204358905164 -11.2007144226305 -14.9200270625871  -7.9922802460666
## [909]  -8.3520070119999  -3.0634230377443  -8.6667574666339 -17.3946315863072
## [913] -13.9751881524778  -7.4284136063703 -12.8017391914418   0.0796759211016
## [917]   1.5521962402319 -12.0279494913341 -13.2140954708239 -26.2105391706051
## [921]  -4.3189474100682  -3.2871975282205   4.6025952177705 -26.3771937428622
## [925]   2.9850108220997 -11.5369982879414   5.7914884558458  10.5523518552608
## [929]   2.8448665087213  -5.8259668272002   1.5985360728644  -5.0840974428155
## [933]   1.0200871771494  -4.3142956971968 -13.6391708511682  -8.6304362923845
## [937]  -2.7572457289469   5.3998160594704  -0.1300766528528 -19.4602199796870
```

```
## [941] -16.1268068422097   2.8688659034770 -15.5665015992992 -10.2297772216073
## [945]   3.2474112266228  -7.3610233098449  -1.5866098092116 -12.6413748901856
## [949]  -1.1323070809281  -7.7451424430973  -4.5516727533891   1.7625072675557
## [953] -13.1357524185268  -4.9462331148863   5.3970129858793   4.9100271069654
## [957]  -7.5631051533232 -14.1531432635225 -30.3517276221818 -16.4335789783104
## [961]  -9.4617851897070 -13.8956188390521   8.7782555336490 -24.5760672674387
## [965]  -2.7614462162304 -22.1807626153914 -13.0575886988837  -9.7560228337793
## [969]  -6.4773389881232  -1.7621024448896  -4.0221343890048   2.9563746444499
## [973] -15.7806617916108  -8.2702501953435 -15.8119624600776 -17.8259059633484
## [977] -26.6682920426637 -17.1347324830373 -15.0667582780123 -16.9518394411486
## [981]  -4.5839218883509 -15.8107729636208  -5.6916794897939  -3.2829881041342
## [985] -14.3350730550368 -17.8098342889453  -5.8370181762388 -18.2162059963196
## [989]  -7.8892410964089  -2.4517683025882   7.8010975268365  -3.3405231809554
## [993] -10.1073381954854 -22.0105460642858 -16.4339778448903  -2.0270768952201
## [997]  -4.7175172561890  -7.1633026119330 -10.1213930489634 -11.9191377747732
```

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

```r
#TO-DO

set.seed(10)
v = as.vector(rnorm(1000, -10, 10))
v
```

```
##    [1]  -9.8125382905817 -11.8425254206906 -23.7133054992251 -15.9916771578372
##    [5]  -7.0545487343249  -6.1020569929983 -22.0807617542949 -13.6367601747086
##    [9] -26.2667268170309 -12.5647839412399   1.0177950308713  -2.4421849197266
##   [13] -12.3823355601872  -0.1255529658661  -2.5860987161618  -9.1065273350418
##   [17] -19.5494385615238 -11.9515038466724  -0.7447873790592  -5.1702147516339
##   [21] -15.9631063672021 -31.8528683816953 -16.7486593787512 -31.1906119191017
##   [25] -22.6519802153090 -13.7366155515470 -16.8755543038792 -18.7215882671769
##   [29] -11.0176100622482 -12.5378053010246 -28.5374045447914 -10.7794606607537
##   [33]  -0.3143365947546  -8.1507404000969 -23.7994357833758 -24.3551436236037
##   [37]  -6.3791277139339 -27.5908675375971 -13.2454400957233 -16.5156298854466
##   [41]   0.8655139944051 -17.6254488003129 -18.2866253500109  -1.6552609691155
##   [45] -19.6765198675991 -10.2881533547586  -7.6747484742461 -13.0120868150418
##   [49] -16.7761458314933  -3.4477236376478 -14.0063754703174 -13.3455656507335
##   [53]   3.6795395319196  11.3776710365012  -4.9418073547097  -2.1365761576084
##   [57] -19.0221194417864  -4.6710300767167 -16.4589425354921  -7.0901251157023
##   [61] -22.3759446887722 -14.5617627511781 -18.3032265472473  -6.5988435632574
##   [65]   0.6637639568217   2.1612583807980  -2.6430934236695 -14.8120861731558
##   [69]  -4.3725523714188 -22.4631971188920  -6.1907778737432 -24.3042725279669
##   [73] -20.4844550487860 -12.1850355053459 -24.8993623673554   1.7270628121431
##   [77] -24.7982702157166 -14.3038781607722 -20.5163864204345   5.2258634405410
##   [81]  -4.0717194541392 -12.2266150901927  -2.8710572375154  -2.8339916625895
##   [85]  -5.5975813561586  -8.4116937868190  -3.4023586166805  12.2051966293556
##   [89] -21.8394507406540 -10.7395583449749 -14.1635467488652 -11.9148234375263
##   [93]  -9.3045521859260   1.5534831801052  -4.0504265304951 -24.1964510835699
##   [97] -26.0667724535804  -1.0707410043682  -8.5183204481227   2.2702839010139
##  [101] -17.6180433917803  -5.8062459411009 -20.3994336463235  -2.8842603400737
##  [105] -16.3321301496783  -4.3682533554985  -3.3901331416840 -26.5805085732545
##  [109]   0.2816797701792   1.2795361401459 -22.8015460342218   1.2886822740957
##  [113] -14.6413452716498 -13.1576020953137  -0.7570685316505  -9.2285527601422
```

```
## [117]   0.3992360511188  -2.5811379326182   2.5554485828951  -0.4908103354382
## [121] -14.8136560727329  -7.9711822203016 -10.3173974383773 -21.9558030033457
## [125]  -3.7631876315157 -19.1480448366691  -7.5124199229190 -20.6262279318038
## [129] -13.6398224719576 -22.0699485337827   4.2921278138977  -3.6656410901717
## [133] -29.9681561765642 -16.8183217309621 -14.6005547931070 -19.8306919414776
## [137]  -5.0466828711166  -2.7418249976747  -3.3270126810708  -0.4521356353341
## [141] -26.7533217929194 -22.0518539249191 -29.6325248922053   4.7075230981397
## [145]  -6.2752766144940   0.6587933403768  -4.6935013164268  -8.9801655411587
## [149]   3.3778246578648  -9.1276523150887 -13.9110420740045 -12.4986748459556
## [153]   1.5510474589615 -18.6472723983100 -18.6667834236831 -33.2101703034786
## [157]  -3.9116983108215   1.5000604820099 -21.9959767173686 -25.8000075454988
## [161]  -3.4683380635781 -15.4940848512297  -4.7894547468747 -16.9940306640950
## [165] -14.3890931479220 -16.7731929611669  -0.4085880522465 -24.6817332946240
## [169]  -8.1623610733187 -24.3514718265180 -21.3739989812347 -14.1464532687800
## [173]  -8.5606571184864   0.6202433069828 -15.7079390273540   2.7718137641215
## [177]  -7.7171067945992 -13.0881306450274  -0.4017086940442  -4.5117762518792
## [181]  -5.7448690622802  -3.5649996487183 -23.6030614350351 -11.9850610640319
## [185]  -3.8069732312462  10.6820960503683 -13.0528475419725  -7.1875438777014
## [189]  -3.0868266329410  -9.5363856185668  -8.8697063837404  -0.0466812571087
## [193] -16.8115136132165 -22.7705724668769 -24.6869774983108 -13.1347406653587
## [197] -27.0365949266331 -23.5051465613771 -21.0209367720916 -20.9954301452935
## [201]   2.1551377619565  -6.6912351489586   3.9027511927468  -1.2795301497889
## [205] -20.8081702266027  -5.0417840855346   0.5262755621631 -22.7464995015128
## [209] -11.9366672842338 -22.9508364255337  -8.5811971532135   2.6171505198662
## [213] -14.3150032806965 -28.2271258950182  -6.4745604124689 -23.4845144147378
## [217]  -2.9231168209257 -14.1089093679878 -14.4604517992225 -20.4115630166875
## [221] -13.2922471615401 -12.8282162413235  -5.6757087450465 -13.0760709506342
## [225] -10.5663630788932  -2.6648457979973  -9.0268838033482   6.3089173658816
## [229]  -4.3938930099780   3.2956476445216 -12.7882349563446 -22.6673154414092
## [233] -12.4914839404027  -9.8201158543120  -6.2292727288967  -2.0399144247256
## [237] -18.4067741709857 -32.0547175082621 -21.2805599106249 -23.4130995880245
## [241]   6.0511404288097  -2.5557627151146  -1.3791779404662  -6.0484418452380
## [245]  -4.9088130366967 -11.2255012583063  -9.0741535282172 -13.5787991433363
## [249] -13.5965522434701   0.2857072088428   0.7789259245098  -0.6821878476750
## [253] -24.6079386697760 -19.0607557624934 -16.8034783139241   0.6316603678904
## [257] -16.9241315212615 -21.3362826529090 -20.9461543379722 -20.1290361883673
## [261]  -5.8972267411411  -5.2225955807321 -33.2987954506654  -9.8374453482326
## [265]  -0.1964647685352  -1.9365173518264  -8.8039911110870 -34.3661560431758
## [269]  -9.6897752387761 -13.3574669799984 -12.6377616192376 -13.5585246088020
## [273] -14.8115479829018  -7.7726453218213  14.2991031869739   4.9629811873421
## [277] -17.1720820275909 -14.6705390018763  -3.3707654714059  13.0017689094660
## [281]  -6.7248985878341  -9.3613651838309 -21.3956093996767   1.8041020362035
## [285]  -9.5861152296685 -22.1359356338470  -9.2680422337219 -12.5732459434554
## [289]  -7.3319358905787   3.8772432571642  -8.0692036497606  -4.0768337739957
## [293] -18.2997452691578  -6.0742671835751  -6.1513238931919   0.5104466572565
## [297]   1.5579748458375 -20.3443795894203 -12.5446806580427   2.7368426608477
## [301]   5.0254463898139  -4.0959053285336 -16.3068545138345  -2.0765046270573
## [305]  -8.7461542987692  -6.7724502875197 -14.4558320945259  -2.3315606553862
## [309] -24.0350302182772 -21.7604677961141  -4.8840353081520   3.1676526377830
## [313]  13.9291294709557 -10.6776940849240 -12.0378506645272   9.9643445216969
## [317]   5.4321779434101 -22.2833762626182  16.4448799222478  -2.4298629172563
## [321]   0.5534519215293  -0.4205510469419  -1.6612094235735 -11.6927557001864
## [325] -14.0443364307844  -7.0439842629709 -12.3187450723486 -14.4807823162639
## [329]  -9.7768920342761  -9.5553270276387  -5.1927883470936  -3.6438014483043
```

```
## [333] -30.0074239307650 -16.9065550376296 -11.3064518930652 -13.4425178302364
## [337] -10.5241282396513  -9.0387607945088  -7.3359294676787  -4.4520652472126
## [341]   2.3544587854851  -7.1486661869311 -14.5879292777288  -3.7652197813763
## [345] -17.2399509164038   6.1941870348572 -16.1667341021694  -6.1545894255312
## [349]  11.1452131280045 -20.2856736628788 -18.8678796566786   2.7114600133104
## [353] -26.0508544127984   1.2227338293318  11.5843864176903  -5.7175338697736
## [357]   2.0117869045970   0.3169011615184  -3.4612574728974  10.1208177317955
## [361]   0.9462984183357  -2.7359064591801 -23.9965713658319 -22.8779516329080
## [365] -19.6841547752596 -27.3668575771840  -1.6830264192646 -17.5895310856506
## [369]  11.5507700106696 -14.2325888023502 -10.0564993667530  -5.9963827138848
## [373] -13.6567780204369 -14.2914383522530  -7.5530287320270  12.1977624873515
## [377]  -6.8573019208592 -15.6378507451632 -24.8569615831883  -5.8470140690619
## [381] -15.2806368267859 -13.5047061745097  -2.6937806270378 -15.5662416507982
## [385] -24.4976053616599 -20.5199642811248 -16.8283861878505 -16.1294117857592
## [389]   8.9078003505454 -11.4541053359420 -16.3942372058860 -14.1117604522169
## [393]  -6.5720812156793 -21.3324675462843  -9.3392969748082  -9.6200023403604
## [397]  -0.7893525245555  11.3496563845849 -15.9982693155619   2.4444645863646
## [401] -10.7945701712045   1.8175154981561  11.8614406267608  -5.9382506929464
## [405] -17.3835910458486 -29.5648670291891 -29.5004570159855 -19.4099781569260
## [409]   1.9778676862693 -16.2423939035829 -11.3286965510209 -10.1722092219131
## [413] -14.5962928217906   4.7293679052688  11.6931961261593 -40.0143130631433
## [417] -27.7198558305170 -13.6483922283578  -6.2546621931091 -22.3407792667110
## [421]  -5.2510657804091   2.3305745705665 -15.9312865679849  -2.4773756067248
## [425]  -3.8795449311322 -12.2865736527439 -17.0483934989938   2.2951604720639
## [429]   5.1851512986719   3.4509418039941 -11.2369152893043  -0.6609754644998
## [433]  -9.8050315437978  -1.6914789025703 -13.0449466799607 -10.5928127710359
## [437]   0.0739669737882 -23.7779508602292 -20.0420272696930 -10.2328030491504
## [441]   5.4674727378832 -17.5936325102132   0.0679265745540  -7.2578508365171
## [445] -10.8038236456008 -19.0355249532682  -5.2685489702804   3.8848897790595
## [449]  -9.6525390623583 -22.9258502807925   1.6396750490781 -25.2338459706839
## [453] -35.1833505361223 -17.0752923056637 -12.8743288270911 -14.3530831147691
## [457] -13.4307877074330 -10.3931088400990  -1.1303401227621 -13.5994840081589
## [461] -17.2895427063139   7.5174561927189 -10.8603859023211   1.0636301587614
## [465] -10.1917783723957 -24.9346017845903  17.0063662306933 -17.4369442224344
## [469]   0.6519498093684 -10.3208173166496 -15.3995968933919 -29.2564356755388
## [473]   0.0783137615909 -12.7923479676676 -22.4956585528460   2.4849112786506
## [477] -11.9221337656587  -0.8802376891256 -31.5826968002626   2.8961053233286
## [481] -14.2810996555383 -12.5500289297295   3.2715823880373  -1.6130987938999
## [485] -24.6500067717319 -16.1625269787270  -8.9389416059358 -28.4401675772715
## [489] -26.1642285371569  -7.7232329586370  -0.7878329806117 -13.3120343949900
## [493]   2.8723403098892  -9.4273919754899 -32.0393312516692 -15.2291034826556
## [497]   0.8078531012050 -12.4314133761565 -19.0944656225310 -18.6829950737599
## [501]  -1.3052495509230 -16.8000959781380  -8.2678546129149 -11.5943803907549
## [505]  -2.0650056735672   6.9435049142761   2.3996868635003  -9.7056325685223
## [509]   6.5721449904792   1.3122314018687 -24.0241063212584  -8.4322953888707
## [513]  -1.3249140356862 -10.3323515710174  -0.4593544367399   1.3009672517014
## [517]  -6.0630268791701  -6.2319167203290 -12.5174290704864  -4.9590510917322
## [521] -23.2980347134830 -11.1829945251477 -16.3324846660885  -3.2230865528305
## [525] -12.9788176477430  -3.7207775287270   6.7843072799152 -34.0000254062929
## [529] -23.4984288507140   2.0200789578675   2.9384271937366 -10.5959434618245
## [533] -16.8506234517232 -24.5343930439156   2.9412656120527   1.7616908935200
## [537]  -1.3595492182401 -32.9673252346068 -23.9181001193225 -23.8587674502157
## [541] -20.7304684391546 -19.4142550319634 -15.8643430972218   0.3472122005478
## [545]  -5.4103779330522 -21.9344507199800  -2.0811072832840 -27.7230491079952
```

```
## [549] -15.8772242199909 -22.8237031255411 -13.1213466098008 -16.9877530062472
## [553] -36.4384976770901  -1.7497762320080  -6.9307368678959 -10.5803945817092
## [557]  -0.7343058916559 -12.1869186057380  -0.0686650257996 -29.2333434248037
## [561] -20.5271518447267  -1.0960223807401  25.4114027762577  -2.5848925731212
## [565] -12.0747125409922  -8.2006154580829   4.2636484940947 -14.4316678948262
## [569]  -6.3281001334818  -3.0417830388934  -0.6655527138822   4.9188908255981
## [573]   1.5952299023330  -7.7320404309088 -22.9546518474018 -10.6277869169191
## [577]  -0.0671706312629 -20.0593093995471 -17.5098494759810   7.4401638235168
## [581] -21.8760009145738 -14.1697041771968 -25.6982015936441 -12.8047403525336
## [585] -15.8532696388542  -3.3968080243666 -16.3751254898457 -15.3902163321128
## [589]  -0.4847976644570 -11.0557984584728 -13.7406927276919 -11.7623258082391
## [593]  -7.2351669207907  -7.6366740733879  10.6167415943736 -22.9005998583527
## [597]  -4.2881055252591   7.8539022670878  -7.8705975690789  -1.1530785394932
## [601] -19.0197026441077 -10.5474995804949 -23.5675471109976  -6.8276942287425
## [605]   3.9571911930986 -22.1122381679601 -12.0414166079742 -25.5245132607939
## [609]  -4.8195033138507 -14.1884030323838 -27.0945321567921  14.8316267137122
## [613]  -5.0196626493359 -16.7386820519833 -18.8698139561134 -27.3137735086099
## [617]  -3.9545087863572 -18.7127507822903  -2.7465835362727 -10.3073198012791
## [621] -11.1722237431415  -9.0177706141557 -16.7685722184453 -10.4924635492814
## [625] -15.6071975993502  13.9915486524696 -10.8987210159804 -18.2164841778561
## [629] -12.8696176159712  -8.6872112602923 -20.6208395043809  -2.6641896427090
## [633]  -6.7730780152115 -13.4897977134871   2.8352566424672 -20.1448804888177
## [637] -15.1006427410707 -11.6716974615534 -13.0266778560616  10.2647047246020
## [641]  -0.9909436158995  -8.6457095954416 -12.0128895419938 -18.0683674099314
## [645]  -3.5587587624079 -13.3665912065361 -27.3170351124497  -7.3807933739103
## [649] -19.1669759810932 -15.6856447514912 -18.6582044175357  -1.7450608820195
## [653] -18.0013228420733  -0.2416991321562  17.0007553074187 -11.3538959688570
## [657] -11.0603116395947 -23.4406380514536  -5.5687817138720  -0.3888225833660
## [661] -12.4143581665569  -9.1704510889171 -13.6172475776824  -6.6613980019877
## [665]  -7.3439057902948  20.1290574660275 -13.4473013768743   0.6255745328506
## [669] -17.2968507404050  -9.4345227902359  -5.3217028807995 -13.7264525055132
## [673]  -7.8983831230669   2.8960830567623 -26.2683954986708 -17.8353650968338
## [677]  -4.8908689654305   2.1762605593289  -6.9489075231166  -7.8970636667506
## [681]  -5.3113403449412 -29.0434257226243   2.3840228464858  -6.0900110854235
## [685] -10.3119891674686 -26.1880052176578  -5.4873372186718 -16.9084336362306
## [689]   4.0544562193991 -27.0134268782504  -8.0775440272286 -17.0295676149723
## [693] -18.0855586334880 -16.3819874152539   7.3734542796118 -32.6047826314218
## [697] -26.1247209388646  -3.4625980119919  -0.9711927140699  -2.4539151216549
## [701] -12.2681076877007 -16.9596862759163  -4.9335264266620  -3.8376570639239
## [705] -26.6687927451536 -13.3625422574265  -4.9282014325343 -26.9739070150845
## [709] -12.0692722390273 -13.9859702817616 -21.8861982748089  -5.4743459586091
## [713]  -7.9241029607058 -22.1012237611841  -2.3258494571936 -15.9902769125696
## [717] -25.6097277280565 -19.9566404292894   0.9325100838442  -0.8285711399263
## [721] -17.7304935513235 -14.7583581764580 -21.3070318401985 -14.2883543166679
## [725] -18.3958364986366 -16.0098667290394 -17.7389077819077 -18.9249235117396
## [729] -10.3573184705772 -13.0417810239351 -40.1216378355869 -30.0852305776173
## [733] -10.4637140944499 -13.5953706450041 -11.9502166174765 -25.1447013546087
## [737] -18.9203304712062 -18.6270649141407  -4.4804585204538 -11.8823391331742
## [741] -13.2514674362955 -15.3693844913960 -13.7147741274935  -2.7719297634096
## [745]   2.0569188339316   1.8415312566266  -3.2880956459256   1.7550207341849
## [749]  -8.1147352727879  -5.0093640315497 -29.4761831141199  -5.7926275382354
## [753]   1.7347624977911 -24.4623155699525 -13.6675120594831 -16.0779623487786
## [757]  -6.8012862261922   0.9984767220601 -14.0603662759087  14.1074642682032
## [761]   1.5889988480835  -6.6161040844377  -5.4382265607910 -13.1404466641558
```

```
## [765]   4.0434402019996 -14.1026994908487 -10.8810251615009  -6.7512440528369
## [769]   1.8509980019251 -12.5550857344088  -1.1555207910437 -14.8856106962647
## [773] -17.2737617317756 -21.8825527291670 -15.3794241591258  -8.8064878268304
## [777]   3.1817809545599   4.4308055765009  -7.5097378835476  -1.8429767314940
## [781] -17.3460780637786  -7.4184672197781 -18.5286076882271  -7.8435124144653
## [785] -17.0921021200240  -2.8913447684106 -14.4737185876207 -13.3829830419350
## [789]  -4.8276657785930  -9.0129640567598  -2.5631438416316 -12.7855014148461
## [793]  -4.6092831938205 -10.8833946967005 -17.4575750815326 -25.6007845110139
## [797]   0.8295652542379 -33.4105600256368  -7.9369473396797  -2.3525742881898
## [801] -31.2506858104394  -4.7882133369683 -10.4642289995611 -21.4947172097093
## [805]  -3.7066169229980 -17.2647147599165 -13.8641732230971 -16.7171438988431
## [809]  -0.9961380963144  -1.1315830938367  -6.0972188810054 -19.2751149022227
## [813]  -2.5416077121940  -5.0517584335736 -10.2193146172307  -4.1197767373419
## [817]  -2.4350892599763 -14.9402627261870 -17.3331562861347 -10.0102039398779
## [821]  -1.9417430322349   1.0930938187503 -13.0967948215953 -19.0857824812747
## [825]  -7.3697387231332 -10.2091780757642 -12.1234250764778 -20.3110391977497
## [829]  -8.6048926030375  -8.3378201538353 -19.8465964905627  -9.7736592922938
## [833]  -2.1466996733904 -21.7918952425080   9.1079077756379 -15.3800149491958
## [837]  -1.1572357097661  -8.5269036329116 -20.0842736285689  -1.6181100779241
## [841] -21.6823913740479  -5.4568186708000  -7.8742713249553 -15.5308634742956
## [845]  -9.7656997482033  -3.5182110960861 -27.4279473520386  -4.0483733908813
## [849] -18.7852901298319  -3.5556242870789 -16.2909850266660 -10.9820018968384
## [853]  -3.1578230083224 -23.8468603099041   5.5494558598051 -18.1434134188457
## [857] -18.4363108589257   0.2492742587361  -8.8429662360384  -7.5850596012720
## [861] -23.9081614168685   5.2606084457092  -9.5348376327860   2.3998235965392
## [865]  -6.9142736453721  -2.1437239549793   1.1747870473614 -10.0498227234439
## [869] -11.7841821019019  -3.0377093105292  -4.7820279150645 -14.4782659024346
## [873]  -5.9459939595531 -17.3696442466941  -7.4425198558953 -12.8493350493357
## [877] -10.3112132499334 -14.8585888696012 -11.3508808415585 -24.8071872283751
## [881]   2.5636912568560  -7.9095835009043  -6.8684321040987   0.7514823866383
## [885] -12.8662517457042 -10.3197742743150 -17.4273339663881  10.8522019215140
## [889]  -1.4510958959000   0.0044599614444 -21.8411791008532 -25.4110255512378
## [893] -12.1173300076082  -8.2964743003733 -16.9077737443293   6.9772073903203
## [897] -10.7268996260587 -17.1760885628929 -14.1651443830786  11.9358676105577
## [901]  -4.5765217421848   0.2366760898180  -1.8717389128953   3.0102707741098
## [905]  -4.3204358905164 -11.2007144226305 -14.9200270625871  -7.9922802460666
## [909]  -8.3520070119999  -3.0634230377443  -8.6667574666339 -17.3946315863072
## [913] -13.9751881524778  -7.4284136063703 -12.8017391914418   0.0796759211016
## [917]   1.5521962402319 -12.0279494913341 -13.2140954708239 -26.2105391706051
## [921]  -4.3189474100682  -3.2871975282205   4.6025952177705 -26.3771937428622
## [925]   2.9850108220997 -11.5369982879414   5.7914884558458  10.5523518552608
## [929]   2.8448665087213  -5.8259668272002   1.5985360728644  -5.0840974428155
## [933]   1.0200871771494  -4.3142956971968 -13.6391708511682  -8.6304362923845
## [937]  -2.7572457289469   5.3998160594704  -0.1300766528528 -19.4602199796870
## [941] -16.1268068422097   2.8688659034770 -15.5665015992992 -10.2297772216073
## [945]   3.2474112266228  -7.3610233098449  -1.5866098092116 -12.6413748901856
## [949]  -1.1323070809281  -7.7451424430973  -4.5516727533891   1.7625072675557
## [953] -13.1357524185268  -4.9462331148863   5.3970129858793   4.9100271069654
## [957]  -7.5631051533232 -14.1531432635225 -30.3517276221818 -16.4335789783104
## [961]  -9.4617851897070 -13.8956188390521   8.7782555336490 -24.5760672674387
## [965]  -2.7614462162304 -22.1807626153914 -13.0575886988837  -9.7560228337793
## [969]  -6.4773389881232  -1.7621024448896  -4.0221343890048   2.9563746444499
## [973] -15.7806617916108  -8.2702501953435 -15.8119624600776 -17.8259059633484
## [977] -26.6682920426637 -17.1347324830373 -15.0667582780123 -16.9518394411486
```

```
##  [981]  -4.5839218883509 -15.8107729636208  -5.6916794897939  -3.2829881041342
##  [985] -14.3350730550368 -17.8098342889453  -5.8370181762388 -18.2162059963196
##  [989]  -7.8892410964089  -2.4517683025882   7.8010975268365  -3.3405231809554
##  [993] -10.1073381954854 -22.0105460642858 -16.4339778448903  -2.0270768952201
##  [997]  -4.7175172561890  -7.1633026119330 -10.1213930489634 -11.9191377747732
```

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

```
#average <- mean

average <- function(x) sum(x)/length(x)
standard_error <- function(x) sd(x)/sqrt(length(x))

Av= average(v)
Av
```

```
## [1] -9.8862525833
```

```
standard_error(v)
```

```
## [1] 0.31364801174
```

- 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?

```
#TO-DO
quantile(v, 0.05)
```

```
##           5%
## -26.189131915
```

- 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?

```
#TO-DO


print("Theoreticall percentile of ‘v‘")
```

```
## [1] "Theoreticall percentile of ‘v‘"
```

```
z=pnorm(0,-10, 100)
print(z)
```

```
## [1] 0.53982783728
```

```
print("Percentile of ‘v‘ that corresponds to the value ~ 0")
```

```
## [1] "Percentile of ‘v‘ that corresponds to the value ~ 0"
```

```r
for(i in  1:100){
y=  i/100
x= quantile(v, y)
if(x<1 && x>-1) { print(x)}

}
```

```
##          81%
## -0.82111960301
##          82%
## -0.48021788347
##          83%
## -0.12930762607
##         84%
## 0.23869179684
##         85%
## 0.62104299086
##         86%
## 0.87489344693
```