Lab 1

Adriana Sham Luo

This lab is due 11:59 PM Satuday 2/9/19.

You should have RStudio installed to edit this file. You will write code in places marked "TO-DO" to complete the problems. Some of this will be a pure programming assignment. The tools for the solutions to these problems can be found in the class practice lectures. I want you to use the methods I taught you, not for you to google and come up with whatever works. You won't learn that way.

To "hand in" the homework, you should compile or publish this file into a PDF that includes output of your code. Once it's done, push by the deadline to your repository in a directory called "labs".

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

```
#TO-DO
options(digits = 11)
pi
```

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

- Sum up the first 100 terms of the series 1 + 1/2 + 1/4 + 1/8 + ...

```
#TO-DO
sum(2^seq(0,-99))
```

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

- Find the product of the first 100 terms of `1 * 1/2 * 1/4 * 1/8 * ...`

```
#TO-DO
prod(2^seq(0,-99))
```

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

- Find the product of the first 500 terms of `1 * 1/2 * 1/4 * 1/8 * ...` Answer in English: is this answer correct?

no this is not correct because the sequence aproaches 0 but it does not equal 0.

```
#TO-DO
prod(2^(0:-499))
```

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

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

```
#TO-DO
x=sum(0:499)
prod(1/2^x)
```

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

- Use the left rectangle method to numerically integrate x^2 from 0 to 1 with rectangle size 1e-6.

```
#TO-DO
1e-6*sum(seq(0,1,by=1e-6)^2)
```

```
## [1] 0.33333383333
```

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

```
#TO-DO
sum(sample(0:1, 100, replace = TRUE))/100
```

```
## [1] 0.52
```

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

```
#TO-DO
mean(sample(0:1,500,replace = TRUE,c(.1,.9)))
```

```
## [1] 0.888
```

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

```
#TO-DO
mean(rbinom(1000,1,0.9))
```

```
## [1] 0.895
```

- Use the `strsplit` function and `sample` to put the sentences below in random order.

```
lorem = "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi posuere varius volutpat. Morbi
#TO-DO
paste(paste(sample(unlist(strsplit(lorem,"[.]"))),collapse = "."),".",sep="")
```

```
## [1] " Donec vehicula sagittis nisi non semper. . Integer dapibus mi lectus, eu posuere arcu ultricies
```

```
#sample(unlist(strsplit(lorem, split = ".", fixed = TRUE)))
```

- In class we generated the variable criminality with levels "none", "infraction", "misdimeanor" and "felony". Create a variable `x_2` here with 100 random elements (equally probable) and ensure the proper ordinal ordering.

```
#TO-DO
levels = c("none", "infraction", "misdimeanor", "felony")
x = sample(rep(levels, 25))
x_2 = factor(x, levels = levels, ordered = TRUE)
```

- Convert this variable to binary where 0 is no crime and 1 is any crime. Answer in English: is this the proper binary threshold?

No, this is not the proper binary threshold, I think it should be other way around.

```
#TO-DO
as.numeric(x_2!="none")
```

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

- Convert this variable to an unordered, nominal factor variable.

```
#TO-DO
levels = c("none", "infraction", "misdimeanor", "felony")
x = sample(rep(levels, 25))
x_2 = factor(x, levels = levels, ordered = FALSE)
```

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

```
#TO-DO
p=3
```

```r
n=100
x_3 = ifelse(as.numeric(x_2)==1,1,0)
x_4 = ifelse(as.numeric(x_2)==2,1,0)
x_5 = ifelse(as.numeric(x_2)==3,1,0)
x_6 = c(x_3,x_4,x_5)
Matrix_7 = matrix(x_6,n,p)
Matrix_7
```

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

```
##  [46,]    1    0    0
##  [47,]    0    0    0
##  [48,]    1    0    0
##  [49,]    0    0    1
##  [50,]    0    0    1
##  [51,]    0    0    1
##  [52,]    1    0    0
##  [53,]    0    0    0
##  [54,]    0    0    0
##  [55,]    0    0    0
##  [56,]    0    1    0
##  [57,]    1    0    0
##  [58,]    0    1    0
##  [59,]    0    1    0
##  [60,]    1    0    0
##  [61,]    0    0    1
##  [62,]    0    0    1
##  [63,]    0    0    0
##  [64,]    0    1    0
##  [65,]    0    0    0
##  [66,]    0    1    0
##  [67,]    0    1    0
##  [68,]    0    1    0
##  [69,]    1    0    0
##  [70,]    0    1    0
##  [71,]    0    1    0
##  [72,]    0    0    0
##  [73,]    0    0    0
##  [74,]    1    0    0
##  [75,]    0    0    1
##  [76,]    0    0    0
##  [77,]    0    0    0
##  [78,]    0    0    0
##  [79,]    1    0    0
##  [80,]    0    0    0
##  [81,]    0    0    0
##  [82,]    1    0    0
##  [83,]    0    1    0
##  [84,]    0    0    0
##  [85,]    0    0    1
##  [86,]    0    0    0
##  [87,]    0    0    1
##  [88,]    1    0    0
##  [89,]    1    0    0
##  [90,]    0    0    1
##  [91,]    0    1    0
##  [92,]    1    0    0
##  [93,]    0    0    1
##  [94,]    0    1    0
##  [95,]    0    0    1
##  [96,]    0    0    0
##  [97,]    0    1    0
##  [98,]    1    0    0
##  [99,]    0    0    0
```

```
## [100,]    1    0    0
```

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

Holding on to columns constants in 'Matrix_7', then sum up row by row

```
#TO-DO
sum(Matrix_7[,1])
```

```
## [1] 25
```

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

Holding on to rows constants in 'Matrix_7' then sum up column by column

```
#TO-DO
sum(Matrix_7[1,])
```

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

- 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 24% 1's.

```
#TO-DO
n = 100
Y = c(rnorm(n, mean = 17, sd = sqrt(38)),
runif(n, min = -10, max = 10),
rpois(n, 6),
rexp(n, rate = 9),
rbinom(n,20, 0.12),
rbinom(n,1,.24))
matrix(Y,100,6)
```

```
##                 [,1]           [,2] [,3]             [,4] [,5] [,6]
##   [1,] 24.7205435831 -8.06237841491    6 0.01094408507670    3    0
##   [2,] 22.3116642146 -3.95923196338   10 0.09435272476616    2    0
##   [3,] 18.7131137279 -8.12834872399    5 0.14737612815728    2    0
##   [4,] 16.7999994385  7.71686346736    5 0.12820668395863    0    0
##   [5,] 11.7057747121  4.11348729394    6 0.16901230796199    2    0
##   [6,] 20.4383686653  8.24662493542    8 0.04664889137461    1    0
##   [7,] 19.5025984032 -0.01727274619    5 0.17046257521438    1    1
##   [8,] 18.8812378421 -0.14763327781    6 0.07133281996681    1    0
##   [9,] 15.0885953353 -3.70882652234    2 0.02869619248021    3    0
##  [10,] 15.7656102412 -0.22428234108    9 0.31728674607400    2    0
##  [11,]  8.7252328418 -4.53299746849    7 0.13049672564232    2    0
##  [12,] 24.2285654022  2.53220828250    6 0.02776887830129    3    0
##  [13,] 19.4056877790  7.91730358265    2 0.07926315271177    0    1
##  [14,] 21.1975940348  1.02218757849    7 0.03179488465604    2    0
##  [15,] 24.8175918705 -4.71558948513    8 0.00815019146022    3    0
##  [16,] 29.7192986902  2.42609514855   11 0.37464304928767    3    1
##  [17,]  6.9645251495 -0.88852248620   10 0.15101580931856    0    0
##  [18,]  6.6858749395 -8.96709001623    9 0.00955181515731    3    0
##  [19,] 15.1310829313  8.69104453363    7 0.12897884793527    1    1
##  [20,] 18.6286966264 -5.46451722272    2 0.04091703279940    3    0
##  [21,] 13.2717834394  5.89888566639    9 0.10748202998242    3    0
##  [22,]  8.6500037299 -9.22534415964    3 0.05128092339469    5    1
##  [23,] 25.0073473160  4.19362682384    3 0.16716834045074    2    1
```

```
##  [24,] 25.2674639083 -8.18174490239    1 0.02465085473119    2    1
##  [25,] 15.6594344928  4.99314488843    7 0.14935815716730    3    1
##  [26,] 13.5763958910 -4.18768234551    6 0.37333512387131    4    0
##  [27,] 24.2705564078 -8.33100645803    4 0.06474672739084    0    1
##  [28,] 15.5541957538 -7.13730926625    4 0.06487654428929    3    0
##  [29,] 18.0872542837  1.29646782763    4 0.09191979519142    2    0
##  [30,] 17.2929104773 -0.65263034310    9 0.19814180395564    3    0
##  [31,] 12.0363468779 -3.04453215562    7 0.07250368579601    3    0
##  [32,] 13.1475097849  5.34054160118    7 0.03034160692319    5    0
##  [33,] 18.5123273330  1.92080613226    7 0.07676083309990    1    1
##  [34,] 17.5986660957  3.15058721229    7 0.17125939972741    3    0
##  [35,]  6.1461046440 -0.85594054312    6 0.00089594494137    4    0
##  [36,]  5.4769228410  7.19044561964    1 0.08203752199293    4    0
##  [37,] 21.8713946675  9.16836548131    6 0.30638393737631    3    0
##  [38,] 15.9147527498  9.17990142945    8 0.04697757597185    2    0
##  [39,] 16.1908286236 -4.16006027721    3 0.01526969329764    2    0
##  [40,]  8.1553805760 -1.36561708525    7 0.02131624825092    3    0
##  [41,]  6.8220680219 -4.46886441670    5 0.02490279776976    1    0
##  [42,] 25.4775348565 -3.65066298749    2 0.17812058195311    2    1
##  [43,] 15.7205472585  2.83385762479    5 0.04360223514959    2    0
##  [44,] 16.5901559498 -3.37781768292    2 0.07178985394744    0    0
##  [45,] 11.6219095093 -4.56818677951    3 0.02376114606365    1    1
##  [46,] 20.8392582303 -4.28238237742   10 0.26996524609881    3    0
##  [47,] 20.0836188425 -1.46155992523    6 0.00908147514445    2    1
##  [48,] 13.0190428653  7.04387013335    5 0.35305085031311    1    0
##  [49,] 22.7932234611 -8.59238242265    8 0.17828085686492    1    0
##  [50,] 20.7515897151 -2.52057094593    9 0.36649537746894    4    1
##  [51,] 12.5947568673 -2.80836817808   13 0.03273521808700    3    0
##  [52,] 18.5762275131 -0.45039010700    6 0.05182672328212    2    0
##  [53,] 11.2715496577 -1.54805319849    6 0.07308219776799    0    1
##  [54,] 16.7893947983 -3.66895015351    7 0.04066619866838    1    1
##  [55,]  8.5214488760 -5.33612576313    4 0.20683875720428    4    0
##  [56,] 14.1655208686  9.74980569910    6 0.02878403232708    3    1
##  [57,] 17.9515350100  9.79449629318    3 0.00076757588734    5    0
##  [58,] 21.5077048776  7.91577036027    9 0.03209276031703    1    0
##  [59,] 17.1284041456  2.61803993024    9 0.11018196534313    3    0
##  [60,] 22.7574535285 -2.58733776398    2 0.07979521777623    3    0
##  [61,] 27.8670771918  3.95553861279    5 0.02674440761070    0    1
##  [62,] 22.7452921056 -0.85843795445    3 0.16773466061281    1    0
##  [63,] 26.4815498629 -7.88468139712    9 0.00205469347964    2    0
##  [64,] 18.9669509379  2.63469364028    3 0.03894734263627    1    0
##  [65,] 12.0238469466 -1.05362338480    4 0.06952743558213    2    0
##  [66,] 24.1725372154  6.84775969945   10 0.22934646117085    3    0
##  [67,] 24.4505532220  7.75061437394    3 0.15347391056051    5    0
##  [68,] 23.2152537711 -5.55536497384    4 0.30564475327158    1    0
##  [69,] 16.8171283556 -2.21755626146    1 0.19881446263200    1    0
##  [70,] 21.6000845748  2.75474109687    7 0.04208335451161    1    0
##  [71,] 18.1852173614  0.67001386080    4 0.07417371506906    5    0
##  [72,] 19.7760982093  7.82103755511    3 0.21104707195421    2    0
##  [73,] 25.9345298156 -2.01922473963    3 0.15176874098626    4    1
##  [74,] 10.8523907586  7.53610094544    3 0.20823345373796    3    0
##  [75,]  3.8993400064  2.46438578703    4 0.01375961179535    4    0
##  [76,] 12.0105625256  7.25231354125   10 0.04259102543195    3    1
##  [77,] 19.5643463196  6.77801612765    6 0.24783118750741    4    0
```

```
##  [78,] 19.3211924131  0.25661529042    9 0.05468405239905    1    0
##  [79,] 12.3461999129 -5.19998587202   10 0.14164551771297    2    0
##  [80,] 17.3172848843  4.25595063251    7 0.03019430033035    5    0
##  [81,] 25.8400498250 -3.83801254909    8 0.26324129993144    0    0
##  [82,] 13.7761474352 -6.63758263458    7 0.21819655203051    1    1
##  [83,]  9.2928692790  8.72341793962    4 0.01129343707321    4    0
##  [84,] 21.0146622303  7.10055818316    5 0.01661956144704    4    1
##  [85,]  7.0254884326  7.30243557598    8 0.05536053976458    2    0
##  [86,] 29.8643164548  0.86591174826    5 0.07389942753232    2    1
##  [87,] 21.1604636066  0.38248109631    9 0.07441364942739    4    0
##  [88,] 12.8396043806  9.65444985311    7 0.11825824027600    5    1
##  [89,] 10.4289660211 -7.13922646828    6 0.34591088110991    1    0
##  [90,] 24.7163710002 -0.79917861149    7 0.13121321568541    2    0
##  [91,] 16.5351376090  8.87987470254    5 0.01087242613236    3    0
##  [92,] 11.4354854798 -2.05350559670    5 0.01269567818340    1    1
##  [93,] 28.2359728563 -8.47537295427    6 0.01589842226488    2    1
##  [94,]  7.4327729216  6.47325596306    7 0.04242160907150    0    0
##  [95,] 13.1085510880  9.83522627503    6 0.08374904421691    2    0
##  [96,] 18.7557803952 -8.70937766042    5 0.00835408777412    2    0
##  [97,] -1.3144837726  4.68438422307    5 0.03549844435313    3    0
##  [98,] 12.6078719115 -1.38913358096    5 0.13589517554422    3    1
##  [99,] 16.6812144004 -6.72252262477    9 0.06644569435674    3    0
## [100,] 28.0281994738  0.88179081213    4 0.00625930819660    2    0
```