Lab 1

Burhan Ahmed Hanif

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`.

```
options(digits=11)
pi
```

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

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

```
sum(1/2^(0:99))
```

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

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

```
prod(1/2^(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?

```
prod(1/2^(0:499))
```

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

```
# the expression 1/2^n when n is a large number and is approching infinity then essitially the number i
```

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

```
x = sum(0:499)
x
```

```
## [1] 124750
```

```
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.

```
sum(sample(0:1 , 100 , replace = TRUE))/100
```

```
## [1] 0.49
```

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

```
sum(sample(0:1, 500 , replace = TRUE, c(0.1,0.9))/500)
```

```
## [1] 0.93
```

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

```
sum(rbinom(1000, 1 ,.9)/1000)
```

```
## [1] 0.884
```

- 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

paste(paste(sample(unlist(strsplit(lorem, "[.] "))), collapse = ". "),"." , sep = "")
```

```
## [1] "Aenean nulla ante, iaculis sed vehicula ac, finibus vel arcu. Integer dapibus mi lectus, eu posu
```

- 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.

```
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?

```
as.numeric(x_2!="none")
```

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

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

```
level = c("none", "infraction", "misdimeanor", " felony")
y = sample(level, 100 , replace = TRUE)
x_3 = factor(y , levels = level)
x_3
```

```
##   [1] none        misdimeanor misdimeanor  felony      felony
##   [6] infraction  infraction  misdimeanor infraction  none
##  [11] infraction  infraction   felony     none        none
##  [16] misdimeanor infraction  infraction  infraction  none
##  [21] misdimeanor none         felony     infraction   felony
##  [26] none         felony     none        none        infraction
##  [31]  felony     none         felony     infraction  misdimeanor
##  [36] misdimeanor  felony     infraction   felony     none
##  [41] none        infraction  none        none         felony
##  [46] none        misdimeanor infraction  infraction  infraction
##  [51] infraction  none         felony      felony     none
##  [56] infraction  none        misdimeanor none        none
##  [61] none        none        misdimeanor infraction  infraction
```

```
##  [66] infraction   felony      felony     misdimeanor  felony
##  [71]  felony     none        none        misdimeanor infraction
##  [76] misdimeanor  felony     infraction  infraction  misdimeanor
##  [81] infraction   felony     none        none         felony
##  [86] misdimeanor misdimeanor none        none         felony
##  [91] none        misdimeanor  felony     infraction  infraction
##  [96] infraction  misdimeanor misdimeanor  felony     infraction
## Levels: none infraction misdimeanor  felony
```

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

```
n = 100
p = 3
x_4 = ifelse(as.numeric(x_2)== 1,1,0)
x_5 = ifelse(as.numeric(x_2)== 2,1,0)
x_6 = ifelse(as.numeric(x_2)== 3,1,0)
x_7 = c(x_4,x_5,x_6)
x_8 = matrix(x_7,n,p)
x_8
```

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

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

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

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

```
sum(x_8[,1])
```

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

```
# holding the coloum constant cycleing up the rows
```

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

```
sum(x_8[1,])
```

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

```
#holding the row constant and cycling through the coloums
```

- 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.

```
matrix(c(rnorm( 100 , 17 , sqrt(38)),  runif(100 , -10 , 10) , rpois(100 , 6), rexp( 100, 9),
   rbinom(100, 20 , 0.12 ), rbinom(100,1,.24)),100,6)
```

```
##                [,1]           [,2] [,3]             [,4] [,5] [,6]
##   [1,] 14.1528046787 -1.98408534285    5 0.06065320658187    4    0
##   [2,]  8.8278036717 -3.71526080184    6 0.07670339647060    6    1
##   [3,] 26.6701943933 -8.90577016864    5 0.01310919201378    2    0
##   [4,] 21.9194436654  7.01509650331    4 0.25254868836030    3    1
##   [5,] 26.5150008713  6.56973538920    8 0.06431536195386    1    0
##   [6,] 25.4296219402  8.35827454459    7 0.14555751976815    1    1
##   [7,]  6.0484540090  8.94974174444    5 0.03652444828509    1    0
##   [8,] 25.7350278082  5.30183410272    5 0.09386644579725    4    1
##   [9,] 15.2231974499 -7.98128604423    5 0.06757382676420    0    0
##  [10,] 32.4921129528  6.44842448179    7 0.14486119204784    3    0
##  [11,] 27.3994416455 -6.29245467018    7 0.02701010042801    4    1
##  [12,]  8.5648661523  0.50904054195    5 0.03285531140864    2    0
##  [13,] 18.1296622665 -3.36975909304    7 0.07001361214659    2    1
##  [14,] 16.3409699196  1.71442183200    7 0.06092842973562    3    0
##  [15,] 12.2928350699 -5.48732427880    1 0.02510823185245    6    0
##  [16,] 18.2311909375 -5.81901163794    4 0.41239763524947    1    1
##  [17,] 25.0025318499  7.43845356628    6 0.01193250172461    3    0
##  [18,] 15.2594879961 -7.55334743764    4 0.00071114388465    2    0
##  [19,] 12.2308695985  6.68307261076   11 0.09626468280898    0    1
```

```
##  [20,] 11.8511807425 -7.41100812331    5 0.04705577607577    4    0
##  [21,] 16.5390989042 -1.10869066324    4 0.20520607732959    1    1
##  [22,] 15.4035919325  1.30044237245    2 0.35196376416095    6    0
##  [23,] 20.9216876649 -0.20078650210    2 0.10431217691498    2    0
##  [24,] 21.3063448042 -0.49367892556    8 0.02910847967077    1    0
##  [25,] 19.2242733377 -1.86473793350    7 0.16782777134637    2    0
##  [26,] 19.6015358933  4.56990676001    2 0.02742146969669    4    0
##  [27,] 24.8985680852  0.96239855979    4 0.11393424154103    3    0
##  [28,] 21.1232729581  8.40330694336    8 0.06257820677840    2    0
##  [29,] 31.1310094660 -3.89982063789    5 0.19415458311532    3    0
##  [30,]  5.5161780040 -8.16810480785    6 0.09736386210087    3    0
##  [31,] 20.1333521670 -5.73923359625    6 0.07425198942009    3    0
##  [32,] 14.9773530966 -7.28092014790    5 0.03078399898691    5    0
##  [33,] 17.5248298279 -0.99192672409    8 0.20960446508289    3    0
##  [34,] 22.6426933456  0.85830559023   10 0.03893794269404    1    0
##  [35,]  8.3311079806 -0.27531360742    7 0.25573794190510    2    0
##  [36,] 16.5710389589 -1.44192331005    5 0.01105762666298    2    0
##  [37,] 18.2916503167 -4.25617699511    4 0.01747184185459    2    0
##  [38,] 24.7924925568 -2.83979619388    4 0.03199649214124    2    0
##  [39,]  5.2069648561  7.02488645911    2 0.03591473871388    0    0
##  [40,] 11.5369932181  1.68590275105    3 0.03450662200772    1    0
##  [41,] 26.0372489286  8.70311891660   11 0.09209815050112    3    0
##  [42,]  9.4958110301  1.35754309129    6 0.07901648038325    2    0
##  [43,] 14.7816922523 -8.17460630555    8 0.00587884851605    1    0
##  [44,] 14.9601415246 -1.57262462191    3 0.23648734658384    3    1
##  [45,] 16.3706273022 -1.35421156883    7 0.06583363154075    5    0
##  [46,] 25.3638488696 -7.52532066777    7 0.08652206240790    1    0
##  [47,] 18.1499507798  6.08939698897    7 0.01041173313983    2    1
##  [48,] 14.9263482107 -3.81695224904    8 0.10842074380928    2    0
##  [49,] 15.3936343940 -2.08438247442    4 0.23681055442582    3    0
##  [50,] 10.9099315825 -2.16348917689    2 0.11556026649505    5    0
##  [51,] 11.4914825907 -3.96729850676    4 0.04569544281892    4    1
##  [52,] 13.7849826224 -3.56729087420    1 0.21672309333974    2    0
##  [53,]  8.5416340010 -8.41715627816    4 0.10137642184840    0    0
##  [54,] 11.4531937730  8.80906261504    4 0.01460683464797    0    1
##  [55,] 17.3717009627  9.17295082938    7 0.05449000481181    5    0
##  [56,] 14.0243481848  7.50691625755    8 0.01218151585716    2    0
##  [57,] 22.7862352425 -9.84026247635   10 0.12714850653738    2    0
##  [58,] 11.1276227792 -5.19759436604    2 0.03937960338468    3    0
##  [59,] 19.4039379328  5.30327255372    4 0.05813418437416    4    0
##  [60,] 25.4320394165  8.28948543873    6 0.04113270004554    4    1
##  [61,] 17.9646651101  8.46471481025    7 0.35884091189965    2    1
##  [62,] 14.1810086325  2.49148803763    9 0.23477454963732    5    0
##  [63,]  8.8609046725  3.81198197603    3 0.16304044941955    3    0
##  [64,] 11.2931488862  9.54726451542    4 0.09124000328644    3    0
##  [65,] 10.3051493594 -8.60478281509    8 0.16280717044695    1    1
##  [66,] 20.3026423627  4.28140362259    3 0.09075425296169    4    0
##  [67,] 10.5085248576  7.13945546187    4 0.04721452357868    0    1
##  [68,] 19.8430267527 -3.98020881694    8 0.05273652356118    4    0
##  [69,] 28.8655858872 -7.62463956606   12 0.28974289466365    4    0
##  [70,] 24.4290069712 -5.69237047806    6 0.06486555582119    2    0
##  [71,] 16.7559171532 -9.45257322397    8 0.11792113600453    2    0
##  [72,] 18.5595323226 -4.12052751519    7 0.07911850470169    2    0
##  [73,] 23.1201348188 -4.31396988221    4 0.57787897041318    3    0
```

```
##  [74,] 17.6606287135 -8.20380463731    3 0.12941481524405    2    1
##  [75,] 20.1436392824  6.81626343634    9 0.20838689906624    3    0
##  [76,] 14.4723768687 -7.03211864457    5 0.02388811344281    1    0
##  [77,] 20.1052636373 -4.73603264894    7 0.00691799488763    4    0
##  [78,] 18.4009465938  9.91365222726    6 0.12895058933179    1    0
##  [79,] 17.5465338702 -4.79686132167    5 0.00391284371209    3    0
##  [80,] 17.9677525746 -0.75540164951    8 0.02258929023401    0    0
##  [81,] 28.0468745042  5.84567493759    5 0.03068864800864    4    0
##  [82,] 23.2300051638 -9.89037648309    5 0.04294797337614    3    0
##  [83,] 22.6268130802  1.14760728553    2 0.11282015146428    0    0
##  [84,] 19.0614034150  8.73011107557    5 0.25738349527579    5    0
##  [85,] 17.0900401690  3.23381317779    9 0.00713975023892    4    0
##  [86,] 27.5249727902 -2.67543010879    7 0.24394537858920    3    0
##  [87,] 20.5227831890 -9.54905348830    6 0.04513696751868    4    0
##  [88,] 18.2470844735 -0.16458140686    6 0.16569261632023    2    0
##  [89,]  1.4039615319 -5.60574778356    6 0.01473750437920    3    0
##  [90,] 20.4310312792 -8.69472286664    6 0.00044478289783    1    0
##  [91,] 26.7809140075  8.53722631931    5 0.10707394151577    3    0
##  [92,] 10.6843445133  6.25065603293    7 0.00614432640975    1    0
##  [93,] 15.3906652049 -6.69265788514    3 0.06575630258562    3    0
##  [94,] 10.7188113830  0.78524108976    4 0.11736943261852    3    0
##  [95,] 12.1453179897  7.65519899782    7 0.02932205093547    2    0
##  [96,] 12.4041180502 -9.75261631887   10 0.11934835939305    2    0
##  [97,]  7.0840310785 -8.77703596372    7 0.00027158315832    2    0
##  [98,]  5.1132407776 -8.80396410823    2 0.17668219794823    2    0
##  [99,] 21.1978333764 -7.60163233150    5 0.36756634130863    0    0
## [100,] 14.1149553752 -5.51362012047    5 0.01968849631440    3    0
```