Lab 1

Karen Lopez

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 + ...

```
x=seq(0,-99)
#x
x1 = 2^x
#x1
round(x1,3)
```

```
##   [1] 1.000 0.500 0.250 0.125 0.062 0.031 0.016 0.008 0.004 0.002 0.001
##  [12] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
##  [23] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
##  [34] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
##  [45] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
##  [56] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
##  [67] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
##  [78] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
##  [89] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## [100] 0.000
```

```
#sum(2^x)
sum(x1)
```

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

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

```
x=seq(0,-4)
#x
x1 = 2^x
#x1
prod(x1)
```

```
## [1] 0.0009765625
```

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

This answer is correct but not exact.

```
x=seq(0,-499)
x1 = 2^x
prod(2^x)
```

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

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

```
-log10(2)*sum(0:499)
```

```
## [1] -37553.491959
```

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

```
#sum(height times width)

sum((seq(0,1-1e-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.

```
mean(sample(rep(c(0, 1), 100), 100, replace = TRUE))
```

```
## [1] 0.51
```

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

```
mean(sample(rep(c(rep(1, 9), 0), 50), 500, replace=TRUE))
```

```
## [1] 0.912
```

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

```
rbinom(1, size=1000, prob=0.9) / 1000
```

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

- 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 p
```

- 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.

```
criminality=c("none", "infraction", "misdimeanor", "felony")
class(criminality)
```

```
## [1] "character"
```

```
criminality
```

```
## [1] "none"        "infraction"  "misdimeanor" "felony"
```

```
x_3=(sample(criminality, 100, replace = TRUE))

x_2 = factor(x_3, levels= criminality, ordered = TRUE)
```

```
x_2
```

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

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

```
if (!require("car")){install.packages("car")}
```

```
## Loading required package: car
```

```
## Loading required package: carData
```

```
criminality
```

```
## [1] "none"        "infraction"  "misdimeanor" "felony"
```

```
x_2bin = recode(x_2,"c('none')=0;
       else=1")
x_2bin
```

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

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

```
x_2nominal = factor(x_3, levels=criminality,ordered= FALSE)
x_2nominal
```

```
##   [1] infraction  misdimeanor misdimeanor misdimeanor infraction
##   [6] misdimeanor infraction  none        misdimeanor misdimeanor
##  [11] infraction  felony      misdimeanor misdimeanor infraction
##  [16] infraction  none        infraction  misdimeanor infraction
##  [21] misdimeanor infraction  infraction  infraction  felony
##  [26] felony      none        misdimeanor felony      infraction
##  [31] felony      infraction  none        none        none
```

```
##  [36] none        none        infraction  none        felony
##  [41] infraction  misdimeanor none        none        felony
##  [46] none        felony      infraction  none        infraction
##  [51] felony      none        felony      none        misdimeanor
##  [56] infraction  felony      none        misdimeanor misdimeanor
##  [61] felony      misdimeanor felony      infraction  misdimeanor
##  [66] infraction  none        none        none        infraction
##  [71] felony      misdimeanor none        infraction  felony
##  [76] none        infraction  none        none        infraction
##  [81] none        misdimeanor none        felony      felony
##  [86] none        misdimeanor none        felony      misdimeanor
##  [91] felony      none        none        misdimeanor misdimeanor
##  [96] felony      none        none        felony      felony
## Levels: none infraction misdimeanor felony
```

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

```
x_2a = ifelse(as.numeric(x_2)==1,1,0)
x_2b = ifelse(as.numeric(x_2)==2,1,0)
x_2c = ifelse(as.numeric(x_2)==3,1,0)
x_2num = c(x_2a, x_2b, x_2c)
Matrix_x_2 = matrix(data=x_2num, nrow = 100, ncol = 3)
Matrix_x_2
```

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

```
##  [30,]    0    1    0
##  [31,]    0    0    0
##  [32,]    0    1    0
##  [33,]    1    0    0
##  [34,]    1    0    0
##  [35,]    1    0    0
##  [36,]    1    0    0
##  [37,]    1    0    0
##  [38,]    0    1    0
##  [39,]    1    0    0
##  [40,]    0    0    0
##  [41,]    0    1    0
##  [42,]    0    0    1
##  [43,]    1    0    0
##  [44,]    1    0    0
##  [45,]    0    0    0
##  [46,]    1    0    0
##  [47,]    0    0    0
##  [48,]    0    1    0
##  [49,]    1    0    0
##  [50,]    0    1    0
##  [51,]    0    0    0
##  [52,]    1    0    0
##  [53,]    0    0    0
##  [54,]    1    0    0
##  [55,]    0    0    1
##  [56,]    0    1    0
##  [57,]    0    0    0
##  [58,]    1    0    0
##  [59,]    0    0    1
##  [60,]    0    0    1
##  [61,]    0    0    0
##  [62,]    0    0    1
##  [63,]    0    0    0
##  [64,]    0    1    0
##  [65,]    0    0    1
##  [66,]    0    1    0
##  [67,]    1    0    0
##  [68,]    1    0    0
##  [69,]    1    0    0
##  [70,]    0    1    0
##  [71,]    0    0    0
##  [72,]    0    0    1
##  [73,]    1    0    0
##  [74,]    0    1    0
##  [75,]    0    0    0
##  [76,]    1    0    0
##  [77,]    0    1    0
##  [78,]    1    0    0
##  [79,]    1    0    0
##  [80,]    0    1    0
##  [81,]    1    0    0
##  [82,]    0    0    1
##  [83,]    1    0    0
```

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

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

```
sum(Matrix_x_2[,1])
```

```
## [1] 31
```

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

```
sum(Matrix_x_2[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.

```
n = 100
z = 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(z, 100, 6)
```

```
##                 [,1]            [,2] [,3]             [,4] [,5] [,6]
##   [1,]  8.3826228214 -0.179594806395    3 0.16841091442334    2    0
##   [2,] 23.3340441516  0.386856896803    5 0.07208929102247    2    0
##   [3,]  7.8890429451  3.718085028231    2 0.03924601716507    1    0
##   [4,] 16.7271670617 -9.277618094347    8 0.10628959898267    4    0
##   [5,] 11.2380418516  6.107408367097    6 0.25248786039906    2    0
##   [6,] 15.9100992623 -4.143442730419    5 0.00744143154265    5    1
##   [7,] 29.1508383059 -8.287214054726    8 0.33295327588704    4    0
##   [8,]  4.8409995250  0.546595635824    8 0.09462684247590    2    0
##   [9,] 12.1205789630 -5.559562928975    5 0.21139126456805    3    0
##  [10,] 25.4288969595  4.471341078170    5 0.08665002775123    4    0
##  [11,] 21.9631594809  2.757671051659    4 0.02836900602819    0    0
##  [12,] 13.9813600032 -5.024294047616    3 0.04962848582202    2    1
##  [13,] 18.8739868582  9.761429526843    5 0.22423513708737    3    0
```

```
##  [14,] 24.9405561967  5.987352784723    1 0.10856707290176    4    0
##  [15,] 18.8631792444  6.730707297102    8 0.57080814425360    3    0
##  [16,] 17.2583225498  7.183020915836    8 0.05754778730787    1    0
##  [17,] 25.4173819570  4.702513795346    7 0.06664204224944    1    0
##  [18,] 12.3211173555  9.089266969822    4 0.20712359414227    5    1
##  [19,] 22.2204318780  7.033947282471    8 0.04485592804849    2    0
##  [20,] 20.3252443350 -4.237129590474    6 0.06822971910864    2    0
##  [21,] 22.5460475920 -1.546503142454   10 0.27287259510249    2    1
##  [22,]  7.7922350637  5.165252694860   16 0.06872213890569    2    0
##  [23,] 11.1735800201  3.721105507575    7 0.02836697740066    1    0
##  [24,] 15.4118497723 -6.059708371758    3 0.02633861566169    2    0
##  [25,] 15.8011674868 -0.971356630325    7 0.02807145483351    1    1
##  [26,] 23.2092182908  9.900620514527    6 0.07826030281904    7    0
##  [27,]  9.2319546757  7.183487215079    5 0.04323748333587    3    0
##  [28,] 12.9974972928  4.330429439433    8 0.17553525454335    0    1
##  [29,] 17.1486455995 -6.748040462844    4 0.01226951745856    2    0
##  [30,] 30.9953525162  4.311785381287    8 0.10481644731790    4    0
##  [31,] 15.8346566038 -7.912092227489    5 0.01661845617410    1    0
##  [32,] 16.1957071562 -8.486793660559    7 0.01099863524238    2    0
##  [33,] 17.7188739711 -3.374736998230    6 0.02028100932431    3    0
##  [34,] 12.0296248320 -3.943368145265    7 0.00617126720428    0    0
##  [35,]  4.7151679280 -4.134446387179    5 0.02139201273935    3    1
##  [36,] 13.3634940971  6.645396845415    6 0.55341267049638    0    0
##  [37,] 19.4459981104  6.717109545134    6 0.03812277596444    3    1
##  [38,] 16.0128574531 -3.970323260874    4 0.06042683626422    3    0
##  [39,] 17.8847820584 -8.709278278984    8 0.00954866114383    2    0
##  [40,] 17.8382939214  9.344027289189    5 0.01022290132116    6    1
##  [41,] 16.9078568885 -3.522626277991   10 0.32605355562224    2    0
##  [42,]  5.5505653096 -8.884272300638   10 0.15825141236585    2    0
##  [43,]  2.9497792590 -0.218320777640   10 0.10284209333721    1    0
##  [44,]  8.3123580259  3.666197671555    4 0.17128661853466    1    0
##  [45,] 19.8004729275 -1.046288595535    7 0.02234920583852    1    1
##  [46,] 11.8002617230 -8.067338541150    4 0.04942654709642    2    0
##  [47,] 18.0567446261 -4.832858792506   13 0.00706529705268    3    0
##  [48,] 10.8839581437  8.762123724446    2 0.23268800951796    0    0
##  [49,] 10.9687224791  4.245668132789    3 0.10761105489269    2    1
##  [50,] 25.2095005186 -9.779476709664   10 0.01922389269910    6    0
##  [51,] 23.8258067402 -6.925259339623    6 0.14754713864323    3    0
##  [52,] 16.4605634139  4.387594652362    8 0.12226729863954    4    1
##  [53,] 11.2326262129  4.179700715467    6 0.01604554320996    5    0
##  [54,] 18.5099390606  1.835783715360    6 0.04084545625012    2    0
##  [55,] 23.3521083598  7.122171223164    7 0.04494176080657    3    0
##  [56,] 16.4037744125 -6.345756864175    9 0.08478976889150    2    0
##  [57,] 26.0435252534 -3.690701429732    4 0.29770396255212    1    1
##  [58,] 10.4300075955  7.981024975888    8 0.04261352375357    1    0
##  [59,] 13.5976555730  7.533768611029   11 0.03473419919289    3    0
##  [60,] 18.3258604104 -3.160026520491    7 0.09062606313110    1    0
##  [61,] 16.6119712748 -1.634222026914    4 0.05311827175319    2    1
##  [62,] 26.7748514875 -6.892117597163    9 0.20175282159508    1    0
##  [63,] 19.6703410732 -7.042580200359    6 0.16928765031921    3    0
##  [64,] 15.6464626873 -0.258712721989    8 0.08198894539174    5    0
##  [65,] 20.0732586255  1.399198430590   10 0.04330720711086    3    0
##  [66,] 14.4610825725  2.918356140144    7 0.00332063596903    3    1
##  [67,] 13.5655512748  4.539845865220    4 0.07745776242213    3    0
```

```
##  [68,] 23.1960875432 -1.353944600560   10 0.31333566236883    3    0
##  [69,] 26.9647514852  6.585628259927    4 0.19601717735225    1    0
##  [70,] 15.7129418619  1.059583579190    9 0.03258754276774    3    1
##  [71,] 20.8515213005 -9.903895347379    2 0.23541828057260    5    0
##  [72,] 23.6996934105  2.192493984476    4 0.17737979154959    2    0
##  [73,] 15.5283858596 -4.914216087200    7 0.13707125734899    1    1
##  [74,] 25.0958546104  0.350585235283    8 0.08125422806523    4    0
##  [75,]  6.7060471410 -8.223781907000    3 0.19528780938745    4    1
##  [76,] 11.3497399535 -2.692422177643    6 0.06604193585614    4    0
##  [77,] 16.0134722176 -5.997744351625    4 0.05622740261162    2    0
##  [78,] 24.6866366633  0.276803374290    6 0.21523918118633    2    0
##  [79,] 16.4472313280 -2.550042518415    3 0.04079049423622    2    0
##  [80,]  3.9794670272  2.174467639998    6 0.20152212625945    4    0
##  [81,] 13.0892102186  7.834336119704    6 0.20774978368908    1    0
##  [82,] 23.2105780946  3.694070279598    3 0.06141612445936    2    0
##  [83,] 14.7301035996 -5.034239562228    9 0.13846130096321    3    0
##  [84,] 14.6000625162  0.020017167553    7 0.00241446474595    1    0
##  [85,] 25.4996273812 -4.530179686844    5 0.06283771996904    2    0
##  [86,] 22.6993447034 -2.863445859402   11 0.07585943723097    5    0
##  [87,] 13.3544820798  1.164093189873    8 0.00099793205866    0    0
##  [88,]  9.1398447831  5.847586924210    5 0.34067488701202    0    0
##  [89,] 19.0714682018  4.291504044086    6 0.03493947157294    3    0
##  [90,] 18.3305127464 -3.852149145678    7 0.03667893177702    2    0
##  [91,] 25.4051681741 -2.062786938623    6 0.29111802597268    3    1
##  [92,] 22.5371449064 -7.492599640973    4 0.00019242831816    2    1
##  [93,] 10.1068251750 -7.954488350078    3 0.21548127195796    4    0
##  [94,] 16.4051950767  7.085641967133    7 0.01101173555522    4    1
##  [95,] 17.6044014786  1.707672462799    6 0.01209272190721    4    1
##  [96,] 15.0979455474  0.065786954947    7 0.07834424567082    2    1
##  [97,] 14.2545497739  3.969556386583    4 0.11204371011039    3    0
##  [98,] 20.2205143482 -6.645171442069    6 0.34619878313396    1    1
##  [99,]  7.1204499107 -1.624562535435    6 0.01915260998617    2    0
## [100,] 20.4478632765 -4.492320073768    5 0.23905382879318    1    0
```