

Lab 1

Juan D Astudillo

This lab is due 11:59 PM Saturday 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

```
pi
```

```
## [1] 3.141593
```

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

```
pi
```

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

- Sum up the first 100 terms of the series $1 + 1/2 + 1/4 + 1/8 + \dots$

#TO-DO

```
##x=(1/2)^(0:99)
```

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

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

- Find the product of the first 100 terms of $1 * 1/2 * 1/4 * 1/8 * \dots$

#TO-DO

```
#x=(1/2)^(0:99)
```

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

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

- Find the product of the first 500 terms of $1 * 1/2 * 1/4 * 1/8 * \dots$. Answer in English: is this answer correct? this answer is not correct.

//it is not, but this answer is an aproximatio. Since n of $1/2^n$ is very large the answer will aproximate to infinity.

#TO-DO

```
y=(1/2)^(0:499)
```

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

#TO-DO

```
#we add all the products
```

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

```
x = 124750
```

```
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## remember we dont create vector because is 0 and 1. 50 - 50 probability wHich is standard bernoulli
# create the vector is for setting probabilities
sum(sample(0:1, 100, replace= T ))/100
```

```
## [1] 0.42
```

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

```
#TO-DO
#1:0 same as 0:1

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

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

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

```
#TO-DO
##rbinom(x,size,prob)
sum(rbinom(1000, 1 ,.9))/1000
```

```
## [1] 0.919
```

- 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
#strsplit(lorem, "[.]")
#paste(paste(sample(unlist(strsplit(lorem, "[.] "))), collapse = ". "),".",sep = "")
#sample(strsplit(lorem, "[.] ")[[1]])#[1:10]
paste(paste(sample(strsplit(lorem, "[.] ")[[1]]), collapse = ". "),".",sep = "")
```

```
## [1] "Morbi posuere varius volutpat. Cras suscipit id nibh lacinia elementum. Donec vehicula sagittis
```

- In class we generated the variable criminality with levels “none”, “infraction”, “misdemeanor” 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", "misdemeanor", "felony")
y = sample(rep(levels, 25), replace = F) ##here we dont need to replace
x_2 = factor(y,levels=levels,ordered = T)
class(x_2)
```

```
## [1] "ordered" "factor"
```

```
#x = sample(rep(levels,25)) ##replace = F by default
##x_2 = factor( x ,levels=levels,ordered= TRUE)## check levels
```

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

```
#TO-DO
tmp=as.numeric(x_2!="none")##all 0s = no crime =none
```

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

```
#TO-DO
level = c("none", "infraction", "misdemeanor", "felony")
y = sample(level, 100, replace = TRUE)
x_3 = factor(y, levels = level)
class(x_3)
```

```
## [1] "factor"
```

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

```
#TO-DO
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_7,100,3)## matrix(x, n , p) n = 100, p = 3
x_8
```

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

##	[31,]	0	0	0
##	[32,]	1	0	0
##	[33,]	0	1	0
##	[34,]	0	1	0
##	[35,]	0	0	0
##	[36,]	0	0	0
##	[37,]	0	0	1
##	[38,]	0	0	0
##	[39,]	0	0	1
##	[40,]	0	0	1
##	[41,]	0	1	0
##	[42,]	0	1	0
##	[43,]	1	0	0
##	[44,]	0	1	0
##	[45,]	0	0	0
##	[46,]	0	1	0
##	[47,]	0	0	1
##	[48,]	0	0	1
##	[49,]	1	0	0
##	[50,]	0	1	0
##	[51,]	0	0	0
##	[52,]	0	0	0
##	[53,]	0	0	1
##	[54,]	0	0	1
##	[55,]	1	0	0
##	[56,]	0	1	0
##	[57,]	0	0	0
##	[58,]	0	1	0
##	[59,]	1	0	0
##	[60,]	1	0	0
##	[61,]	0	0	0
##	[62,]	0	0	0
##	[63,]	0	1	0
##	[64,]	1	0	0
##	[65,]	1	0	0
##	[66,]	1	0	0
##	[67,]	1	0	0
##	[68,]	0	0	1
##	[69,]	0	0	1
##	[70,]	0	1	0
##	[71,]	1	0	0
##	[72,]	0	1	0
##	[73,]	0	0	0
##	[74,]	0	1	0
##	[75,]	1	0	0
##	[76,]	1	0	0
##	[77,]	0	0	0
##	[78,]	1	0	0
##	[79,]	0	1	0
##	[80,]	0	0	1
##	[81,]	1	0	0
##	[82,]	0	0	1
##	[83,]	0	0	1
##	[84,]	0	1	0

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

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

```
#TO-DO
sum(x_8[1,])
```

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

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

```
#TO-DO
sum(x_8[,1])
```

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

- 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 is 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
temp = matrix(c(rnorm(100, 17, 38), runif(100, -10, 10), rpois(100, 6), rexp(100,9) , rbinom(100, 20, 0.12)), nrow=100, byrow=TRUE)
temp
```

```
##           [,1]           [,2] [,3]           [,4] [,5] [,6]
## [1,] 25.191465933520 5.299532054923 9 0.0825431157165 5 0
## [2,] -1.488366980553 -4.556347359903 5 0.1057776591383 2 1
## [3,] 29.646288643462 -5.823877234943 6 0.1307712341095 2 1
## [4,] 73.490757309184 9.346363176592 7 0.0227751424744 0 0
## [5,] -3.618424810725 0.268537397496 5 0.1533797744408 2 0
## [6,] 54.582083474477 3.157553914934 10 0.0599870401361 5 0
## [7,] 11.834535659965 3.120759008452 9 0.1165070044244 5 0
## [8,] 26.882911522137 -8.292945339344 6 0.1766780026679 1 1
## [9,] -79.640203340520 8.836102071218 8 0.1386639867966 5 0
## [10,] 48.377734948073 -7.033067825250 8 0.0450202330636 2 1
## [11,] 90.910691004336 -7.789071588777 6 0.0662569735000 2 1
## [12,] -68.893943108274 0.182582433335 7 0.0528978533216 3 1
## [13,] 35.368158298933 3.612252101302 6 0.1780110903819 0 0
## [14,] 32.860830762589 -6.096389400773 10 0.4904829780809 2 0
## [15,] 22.009980428921 1.204814678058 7 0.1511774556665 1 0
## [16,] 14.416134774410 9.373081750236 4 0.0487880210807 3 0
## [17,] -12.890999217479 -8.306430270895 1 0.3340109723897 3 0
```

##	[18,]	23.321480806557	-5.241039809771	5	0.0055057775450	6	0
##	[19,]	-94.975540692284	-1.426947657019	9	0.0963131959727	4	0
##	[20,]	4.223305335078	-1.730220830068	7	0.1472538385701	3	0
##	[21,]	63.234830530873	6.642498853616	5	0.1564522470341	5	1
##	[22,]	5.617178242861	1.772084054537	7	0.3024431319974	1	0
##	[23,]	83.924308283239	9.963380694389	4	0.1220720848870	5	0
##	[24,]	87.663606529969	-9.926382773556	7	0.0793663384346	1	0
##	[25,]	13.161091687106	-9.003569893539	4	0.3483984758805	3	0
##	[26,]	70.125922986129	-7.741102892905	2	0.1844320133649	1	0
##	[27,]	43.832102878403	8.826272077858	7	0.2074119637748	4	1
##	[28,]	62.554270087947	4.249840090051	3	0.0712406296904	1	1
##	[29,]	-3.781608026347	6.759828543290	10	0.1620016029154	0	0
##	[30,]	23.224568457707	-6.380456411280	4	0.1846052170629	2	0
##	[31,]	47.861137856039	5.177980293520	9	0.0374320814717	0	1
##	[32,]	62.209619245084	5.936847999692	5	0.0873780438687	5	1
##	[33,]	84.292392858718	-6.215108265169	5	0.1217181376238	4	0
##	[34,]	39.142277929270	-1.035937531851	7	0.0885842635451	5	0
##	[35,]	-56.671094514994	0.358386109583	3	0.2582002310347	4	1
##	[36,]	25.733962291568	-5.785843548365	11	0.1398332223456	0	0
##	[37,]	29.491988479734	2.671204921789	12	0.0306228863386	2	0
##	[38,]	34.529777578240	0.899161263369	3	0.1004561850910	3	0
##	[39,]	28.686889717921	2.450949507765	6	0.0538249328319	1	0
##	[40,]	27.411066057631	-9.991392353550	6	0.1549874026649	3	0
##	[41,]	-71.721630789045	8.756426875480	8	0.1121466431927	6	0
##	[42,]	20.815020188191	0.587915843353	9	0.1071158502618	1	1
##	[43,]	-65.062602864336	-7.583994809538	7	0.0576449702494	4	0
##	[44,]	16.249264069297	-8.550861920230	8	0.0031133357705	1	0
##	[45,]	-6.524127852097	8.384592942894	10	0.2276470598982	3	0
##	[46,]	37.713248897526	-6.120169973001	5	0.1117351132473	3	0
##	[47,]	57.046512394028	-1.063867444173	9	0.1224560747911	5	0
##	[48,]	13.481933517174	4.426994887181	6	0.0824728792280	3	0
##	[49,]	21.317648411937	-5.445092990994	7	0.0330906795441	3	1
##	[50,]	6.954644774931	-0.438510701060	11	0.1807763336340	5	0
##	[51,]	7.408121677130	4.381077699363	6	0.3451534143170	4	0
##	[52,]	-19.362732104910	8.618607604876	4	0.1958344028558	4	1
##	[53,]	22.561218089279	-5.360303791240	6	0.0446447473433	1	0
##	[54,]	33.859371767732	4.262752262875	11	0.0319031160325	1	0
##	[55,]	12.981546483372	-9.222078695893	8	0.0232447826614	4	1
##	[56,]	-12.996793383572	0.141567969695	8	0.0489491092817	1	0
##	[57,]	-24.858814329178	-3.788981963880	7	0.0942061130532	2	1
##	[58,]	51.096246518988	8.735170951113	8	0.0667869750824	4	0
##	[59,]	-8.776628336017	-2.954955073074	9	0.0833507003216	3	0
##	[60,]	-21.997217826014	6.253320151009	5	0.0276821220090	3	0
##	[61,]	45.502525499082	-3.146302895620	3	0.2758545951985	0	0
##	[62,]	19.760465740608	-5.100255282596	7	0.0623313662493	1	1
##	[63,]	56.285838415442	-5.413582008332	5	0.0021551817398	5	0
##	[64,]	-45.321500898712	-4.916413971223	10	0.0343239427218	2	0
##	[65,]	-28.899869673742	-4.363604066893	12	0.0122100813314	3	1
##	[66,]	-8.146232581671	-0.806690161116	7	0.0428686109371	3	0
##	[67,]	31.042793529915	5.038494081236	4	0.0629639850102	1	0
##	[68,]	10.285915124015	5.756316408515	5	0.1359114241222	0	0
##	[69,]	77.647199099689	-3.063728474081	3	0.0430325654646	3	1
##	[70,]	67.922334675847	-6.479955171235	10	0.0430248863995	2	0
##	[71,]	50.345496055280	2.321961782873	5	0.0203765646244	1	0

##	[72,]	34.892623886086	4.295859262347	5	0.1105219085558	4	0
##	[73,]	-31.215567244446	-4.854225781746	8	0.0792764509103	3	0
##	[74,]	-71.612334930071	6.402284330688	8	0.1519164309480	2	0
##	[75,]	5.604245286068	-3.483237223700	7	0.1679981661246	2	0
##	[76,]	7.654011331462	-2.263132343069	8	0.0410216341766	3	0
##	[77,]	3.122114717919	-3.423047242686	10	0.0089590498764	4	0
##	[78,]	95.871385444146	7.891391864978	2	0.0264875158771	1	0
##	[79,]	-12.488138484413	1.764819533564	5	0.1063935616398	2	0
##	[80,]	-7.304418854587	7.717675068416	3	0.0433362342107	0	0
##	[81,]	-62.792959635779	-0.944987102412	6	0.0453319313626	4	0
##	[82,]	31.104284018842	-7.905853367411	2	0.0274927689073	1	0
##	[83,]	5.631714839822	7.363781812601	4	0.0422490956809	1	0
##	[84,]	51.534599405693	-5.571972727776	4	0.0455005991960	4	0
##	[85,]	63.515231384159	5.082318303175	4	0.1496525224665	3	0
##	[86,]	-14.182431107812	0.074653998017	9	0.0594387164132	1	0
##	[87,]	13.337382035723	-1.997988610528	4	0.0652144852922	4	1
##	[88,]	20.876384840921	-4.109798101708	1	0.0309828626406	6	0
##	[89,]	-49.227214506902	-8.278758712113	2	0.1856780786355	1	0
##	[90,]	41.354642190740	8.469684659503	2	0.0767370303575	2	1
##	[91,]	-58.366165784068	0.386258480139	11	0.0568006116495	2	0
##	[92,]	17.003139233172	9.848246434703	12	0.0381354807566	4	0
##	[93,]	31.375133070649	-1.458611744456	8	0.0755725881706	1	0
##	[94,]	28.448611609778	4.893758292310	7	0.0382591112931	4	1
##	[95,]	26.408395508036	-3.835770748556	5	0.1960260828657	0	0
##	[96,]	-3.621875982416	9.925433397293	5	0.0598602974787	2	1
##	[97,]	-0.081377758453	-0.051802922972	6	0.1123384697730	2	0
##	[98,]	-3.860102944454	-9.723191126250	6	0.0490718062760	3	0
##	[99,]	-30.724300085613	4.482272099704	2	0.1091663671818	2	0
##	[100,]	32.318456361994	1.543366853148	5	0.0387001368735	1	0