---
title: "Lab 1"
author: "Connor Brown"
output: pdf_document

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`.

```{r}
options(digits = 11)
pi
```

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

```{r}
sum(2^(-(0:99)))
```

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

```{r}
prod(2^(-(0:99)))
```

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

```{r}
prod(2^(-(0:499)))
#Yes, if you're multiplying by continuously decreasing fractions, the answer will continuously approach 0.
```

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

```{r}
-log10(2) * sum(0:499)

#The answer is 10E-37553.491959
```

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

```{r}
1E-6 * sum(seq(0, 1 - 1E-6, by = 1E-6)^2)
```

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

```{r}

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

```

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

```{r}

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

```

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

```{r}

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

```

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

```{r}

lorem = "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi posuere varius volutpat. Morbi faucibus ligula id massa ultricies viverra. Donec vehicula sagittis nisi non semper. Donec at tempor erat. Integer dapibus mi lectus, eu posuere arcu ultricies in. Cras suscipit id nibh lacinia elementum. Curabitur est augue, congue eget quam in, scelerisque semper magna. Aenean nulla ante, iaculis sed vehicula ac, finibus vel arcu. Mauris at sodales augue. "
?strsplit
strsplit(lorem, "[.] ")
```
```

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


```

* 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.

```{r}

criminality = c("none", "infraction", "misdemeanor", "felony")
x_1 = sample(criminality, 100, replace = TRUE)
x_1

x_2 = factor(x_1, levels = criminality, ordered = TRUE)
x_2

```

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

```{r}

BINx_2 = factor(x_2, levels = criminality, labels = c("0", "1", "1", "1"), ordered = TRUE)
BINx_2
class(BINx_2)

#Yes, this is the proper binary threshold.
```

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

```{r}

UNOx_2 = factor(x_2, ordered = FALSE)
UNOx_2

```

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

```{r}

inf <-ifelse(x_2 == "infraction", 1, 0)
mis <-ifelse(x_2 == "misdemeanor", 1, 0)
fel <-ifelse(x_2 == "felony", 1, 0)

X = matrix(c(inf, mis, fel), nrow = 100, ncol = 3)
X

```

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

```{r}

#Each row represents a subject. The sum of each row should be 1 if the person has committed an infraction, misdemeanor, or felony. It should be 0 if they are "none".

rowSums(X)

```

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

```{r}

#The column sum will be a 1x3 numeric vector where each column represents the total number of infractions for column 1, the total number of misdemeanors for column 2, and the total number of felonies for column 3.

colSums(X)
class(colSums(X))

```

* 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.

```{r}

col1 = rnorm(100, mean = 17, sd = sqrt(38))
col2 = runif(100, min=-10, max=10)
col3 = rpois(100, 6)
col4 = rexp(100, rate = 9)
col5 = rbinom(100, 20, 0.12)
col6 = sample(c(rep(0,76), rep(1,24)))

X = matrix(c(col1, col2, col3, col4, col5, col6), nrow = 100, ncol = 6)
X

```