Homework 5: Uniform and Normal Distributions, the Central Limit Theorem (CLT)

NAME: Michael Darfler

NETID:mbd25

DUE DATE: October 12, 2019 by 11:59pm

For this homework, it will be helpful to have a copy of the knitted version of this document to answer the questions as much of it is written using mathematical notation that may be difficult to read when the document is not knitted.

Instructions

For this homework:

1. All calculations must be done within your document in code chunks. Provide all intermediate steps.
2. Include any mathematical formulas you are using for a calculation. Surrounding mathematical expresses by dollar signs makes the math look nicer and lets you use a special syntax (called latex) that allows for Greek letters, fractions, etc. Note that this is not R code and therefore should not be put in a code chunk. You can put these immediately before the code chunk where you actually do the calculation.

Some Notation

Your solutions to the problems below must include the formula used for each calculation. To get you started, here is some mathematical expressions written in latex that you may find helpful when writing out the math in your answers. You can copy, paste, and edit these expressions as needed.

For $X \sim N(\mu, \sigma)$ and real numbers a and b:

1. $P(X \leq b) = P(Z \leq (b - \mu)/\sigma)$
2. $P(X \geq a) = P(Z \geq (a - \mu)/\sigma)$
3. $P(a \leq X \leq b) = P((a - \mu)/\sigma \leq Z \leq (b - \mu)/\sigma)$

In this homework we will explore two continuous distributions (uniform and normal) and the Central Limit Theorem. For uniform distribution, probability calculation often reduces to calculating areas of rectangles. For normal distribution, one needs to use z-scores along with normal probability tables (see Appendix B of textbook). Here is a brief review of normal distribution and z-scores.

For $X \sim N(\mu, \sigma)$ and an interval (a, b) on the real line,

$$P(X \in (a, b)) = \int_a^b \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2} dx$$

(i.e., area under the pdf between a and b). As noted in lecture, this cannot be computed in closed form; however, in R this can be computed numerically.

For $X \sim N(\mu, \sigma)$, the probability of getting a value in any interval on the real line can be expressed solely in terms of the cumulative distribution function, $P(X \leq x)$. For any real numbers, a and b, $a < b$:

1. $P(X < b) = P(X \leq b)$, the probability of getting a value less than (or equal to) b
2. $P(X > a) = P(X \geq a) = 1 - P(X < a)$, the probability of getting a value greater than (or equal to) a
3. $P(a < X < b) = P(X < b) - P(X < a)$, the probability of getting a value between a and b
4. $P(X = a) = 0$, the probability of getting the value a

In R, the `pnorm` (x,μ,σ) function is the cumulative probability distribution function for the normal distribution with mean, μ and standard deviation, σ, evaluated at x, i.e. $P(X \leq x)$ = `pnorm` (x,μ,σ) for $X \sim N(\mu, \sigma)$.

Calculating Probabilities Associated with The Normal Distribution Using z-scores

Every normal distribution, $N(\mu, \sigma)$, can be seen as a shifted and scaled standard normal distribution, $N(0, 1)$.

Assume, $X \sim N(\mu, \sigma)$ and $Z \sim N(0, 1)$. Then

$\frac{X-\mu}{\sigma} \sim Z$.

Thus, every quantile in the sample space of X has a corresponding "standardized" quantile in the sample space of Z (called the z-score). If b is an outcome in the sample space of X, the corresponding standardized value of b in the sample space of Z is

$\frac{b-\mu}{\sigma}$ = z-score for b.

Using z-scores, probabilities for X can be determined by transforming each quantile of X into a standardized quantile and using the probability distribution function for the standard normal distribution. For example, for any real b,

$P(X \leq b) = P(Z \leq \frac{b-\mu}{\sigma})$

In R, the `pnorm(x)` function without a mean or standard deviation specified is the cumulative probability distribution function for the standard normal distribution evaluated at x, i.e. $P(Z \leq z)$ = `pnorm(z)` .

Problem 1

Let X denote the checkout time (in minutes) of a customer in a Grocery store. Assume X is a random variable uniformly distributed between 10 and 20 minutes, i.e., $X \sim Unif(10, 20)$.

a. Write down the probability density function (pdf) of X, by replacing the question marks (?) with appropriate values in the next two lines.

$$f(x) = 1 \text{ for } 10 \leq x \leq 20$$

and

$$f(x) = 0 \text{ for } x < 10 \text{ or } x > 20.$$

b. What is the expected checkout time?

15mins

c. What is the first quartile (Q_1) of checkout times? Show your calculations.

The first quartile is (20-10)/4 + 10 = 12.5

d. What is the chance that a randomly selected customer at that store will have to wait more than 15 minutes? Show your calculations.

p(X > 15) X~Unif(10,20) = 0.5

e. Now answer the question in d) using R function `punif` (Type `help(punif)` on the R console to know more about this function). Insert a code chunk below that uses `punif` to calculate this probability. Make sure to set `echo=TRUE` and `eval=TRUE` at the start of your code chunk.

```
punif(15, 10,20)
```

```
## [1] 0.5
```

Problem 2

The daily milk production of a Guernsey cow has a normal distribution with $\mu = 70$ pounds and $\sigma = 13$. A Guernsey cow is chosen at random. Let X = Milk production in one day for a Guernsey cow.

For (a) - (c) answer each question in two ways:

1) Using the `pnorm()` function with the mean and standard deviation for X specified.

2) By converting all probabilities in terms of the standard normal distribution and using the `pnorm()` function without the mean and standard deviation specified.

For both (1) and (2), the formula you are using to calculate each probability must be included before the code chunk where the answer is evaluated.

a. What is the probability that a Guernsey cow chosen at random produces more than 90 pounds of milk in a given day?

$$X \sim N(\mu, \sigma) \quad \frac{X-\mu}{\sigma} \sim N(0,1)$$

$$P(X > 90) = 1 - P(X < 90)$$

$$Z_{score} = \frac{x-\mu}{\sigma}$$

$$Z_{score} = \frac{90-70}{13} = 1.538$$

```
mu <- 70
sig <- 13
x <- 90
z <- (90-mu)/sig

pnorm(90,70,13, lower.tail = F)
```

```
## [1] 0.0619679
```

```
1 - pnorm(z)
```

```
## [1] 0.0619679
```

b. What is the probability that a Guernsey cow chosen at random produces between 85 and 100 pounds of milk in a given day?

$$P(85 < X < 100) = P(X < 100) - P(X < 85)$$

$$Z_{score} = \frac{x-\mu}{\sigma}$$

$$Z_{upper} = \frac{100-70}{13} = 2.308$$

$$Z_{lower} = \frac{85-70}{13} = 1.154$$

```
x85 <- 85
x100 <- 100
z85 = (x85-mu)/sig
z100 = (x100-mu)/sig

pnorm(x100,mu,sig)-pnorm(x85,mu,sig)
```

```
## [1] 0.1137735
```

```
pnorm(z100) - pnorm(z85)
```

```
## [1] 0.1137735
```

c. What is the probability that the quantity of milk produced by a Guernsey cow chosen at random is within 1.5 standard deviations of the mean number of pounds of milk produced by a Guernsey cow?

```
pnorm(mu+1.5*sig,mu,sig) - pnorm(mu-1.5*sig,mu,sig)
```

```
## [1] 0.8663856
```

```
pnorm(1.5)-pnorm(-1.5)
```

```
## [1] 0.8663856
```

Problem 3

As we have seen for other probability calculations, we can simulate from a $N(70, 13)$ distribution to estimate the probabilities computed above in Problem 2.

The `rnorm` (n,μ,σ) function in R will simulate n draws from a normal distribution with mean, μ and standard deviation, σ.

i. Include a code chunk here that simulates 10,000 draws from the N(70, 13) distribution to estimate the probability computed in 2c). As in previous code for estimating probabilities through simulation, you will need to count the number of simulated draws that meet the criterion associated with the event defined in 2c) to be able to estimate this probability.

```
n <- 1000
sample <- rnorm(n,mu,sig)
sum(sample > mu - sig * 1.5 & sample < mu + sig * 1.5)/n
```

```
## [1] 0.868
```

Another Way to Simulate Random Draws from $N(70, 13)$

Randomly drawing from $X \sim N(70, 13)$ using `rnorm(n,70,13)` is equivalent to:

1. Drawing randomly from a N(0,1) distribution
2. Multiplying (1) by the standard deviation of X
3. Adding the mean of X to (2)

So, instead of using `rnorm(n,70,13)` to draw from $X \sim N(70, 13)$, you can also use 13 × `rnorm(n)` + 70 to draw from $N(70, 13)$.

ii. Include a code chunk here that estimates the probability of 2c) through simulating 10,000 draws from the standard normal distribution and using the criterion for counting events associated with 2c) determined in (i).

```
n = 10000
sample <- rnorm(n)
sum(sample > -1.5 & sample < 1.5)/n
```

```
## [1] 0.8668
```

Problem 4

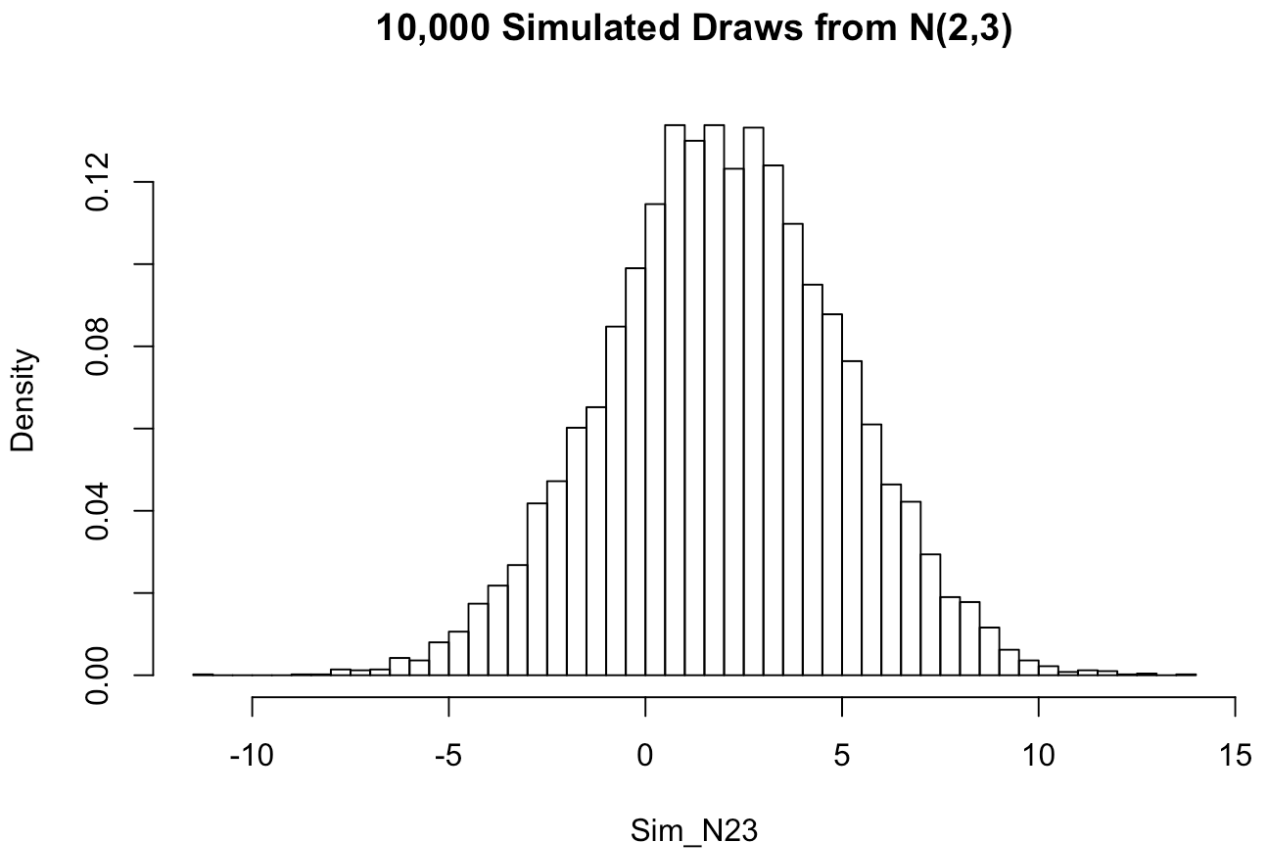
Here we will take a look at how an entire probability density function can be approximated by simulation.

i. In a code chunk here simulate 10,000 draws from a $N(2, 3)$ distribution. Call these simulated values, `Sim_N23` .

```
n <-  10000
Sim_N23 = rnorm(n,2,3)
```

ii. In another code chunk, create a probability histogram of `Sim_N23` using `hist()` . Remember to set `freq=FALSE.` Set `breaks = 75` and `main= '10,000 Simulated Draws from N(2,3)'` .

```
hist(Sim_N23, freq = F, breaks = 75, main='10,000 Simulated Draws from N(2,3)' )
```



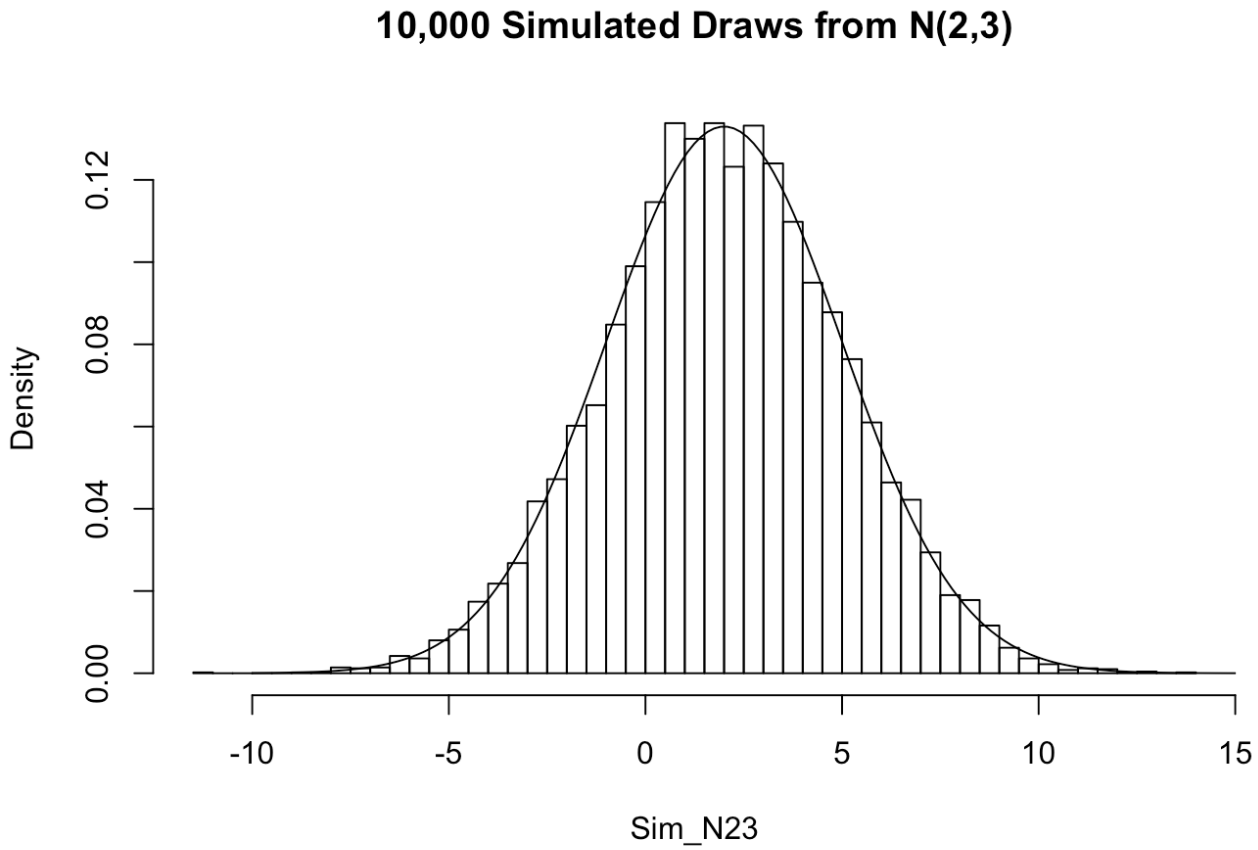
iii. The R function `lines(x,y)` will add a line to an existing plot. The arguments, `x` and `y`, are vectors of x and y coordinates that together define all the points the line must run through. In the code chunk you used to create the histogram above, we will now add code to overlay the histogram with the probability density function for the $N(2, 3)$ distribution. You can do this in three lines of code:

a. Define the x coordinates as `xvalues=seq(-10, 15, length=150)`. `xvalues` will include an evenly spaced grid of x coordinates.

b. Assign the output of `dnorm(xvalues, 2, 3)` to a vector named `yvalues`.

c. Use the `lines()` function to overlay the histogram created in (ii) with the probability density function for $N(2, 3)$. In particular `lines(xvalues, yvalues)` should do the trick.

```
xvalues <- seq(-10, 15, length=150)
yvalues <- dnorm(xvalues, 2, 3)
hist(Sim_N23, freq = F, breaks = 75, main='10,000 Simulated Draws from N(2,3)')
lines(xvalues, yvalues)
```



Problem 5

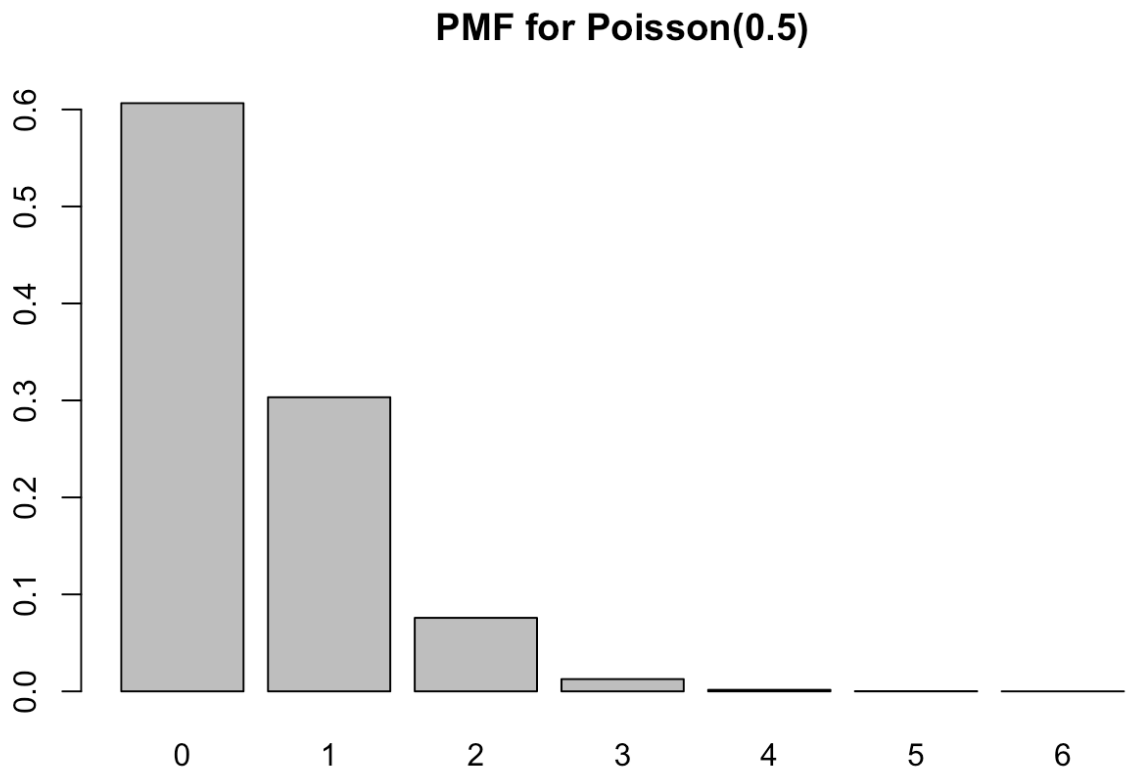
In lecture we saw that the sampling distribution of the sample mean when you draw simple random samples seemed to look like a normal. This is a general phenomenon, known as the Central Limit Theorem. In particular, if $X_i, i = 1, \ldots, n$ is an independent random sample from just about *any* distribution with mean, μ and standard deviation σ, then for large enough n,

$$\bar{X}_n \approx N(\mu, \sigma/\sqrt{n}).$$

Here we will illustrate the Central Limit Theorem by simulating independent draws from the X $\sim$ Poisson(λ = .5) distribution.

a. We start by looking at the PMF of a Poisson(0.5). Change the code from `eval=FALSE` to `eval=TRUE` once you make sure you understand every line of the code chunk.

```
probs = dpois(c(0:6), 0.5)
barplot(probs,names.arg = c(0,1,2,3,4,5,6),main='PMF for Poisson(0.5)')
```



b. Using one or two lines in a code chunk, generate a realization of the sample mean of 4 independent draws from Poisson(.5). You can generate the sample using, `rpois(4,.5)`.

```
sample_mean <- rpois(4, 0.5)
```

c. To simulate draws from the sampling distribution of $\bar{X}_4$, we want to repeat (b) many times, storing the results in a vector.

i. In a code chunk, define the vector of sample means as `xbar.realizations = rep(NA,num.simulations)` where `num.simulations = 10000`.

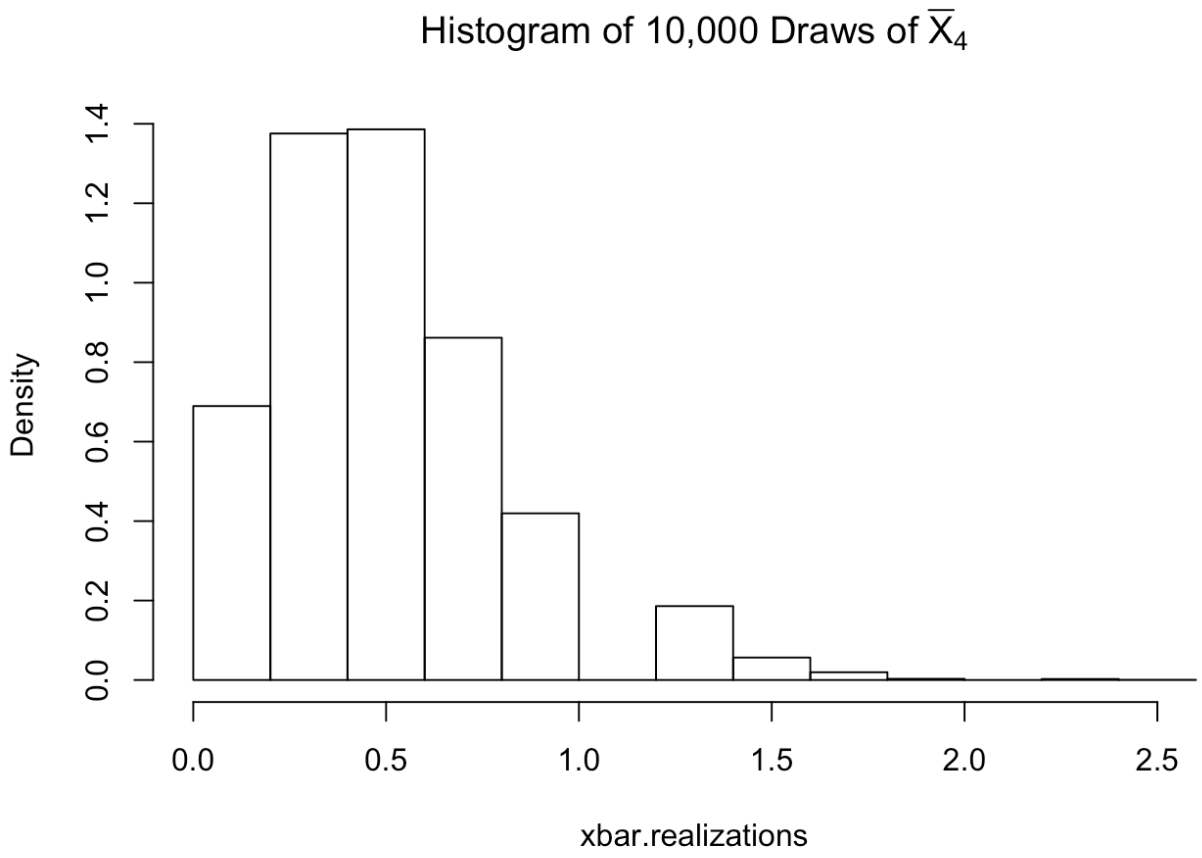
```
num.simulations = 10000
xbar.realizations = rep(NA,num.simulations)
#NANA NANA NANA NANA Batman!
```

ii. In this code chunk, write a `for` loop that repeats part (b) 10,000 times, storing the sample mean generated in iteration i as `xbar.realizations[i]`. `xbar.realizations` will now contain 10,000 draws from the sampling distribution of $\bar{X}_4$.

```
for(i in 1:num.simulations){
  xbar.realizations[i] = mean(rpois(4,0.5))
}
```

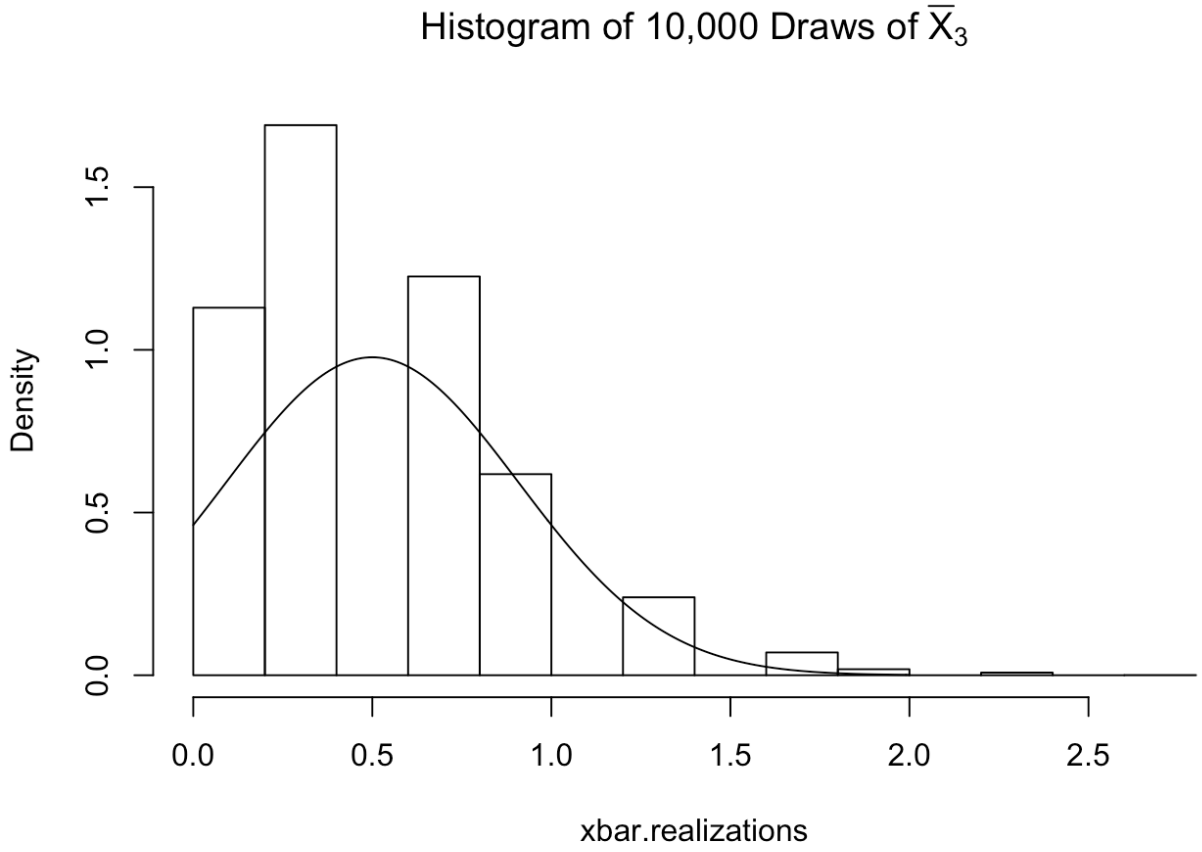
d. Create a probability histogram of `xbar.realizations`. Set `main=expression(paste('Histogram of 10,000 Draws of ',bar(X)[n]))`.

```
hist(xbar.realizations, freq = F, main = expression(paste('Histogram of 10,000 Dr
aws of ',bar(X)[4])))
```



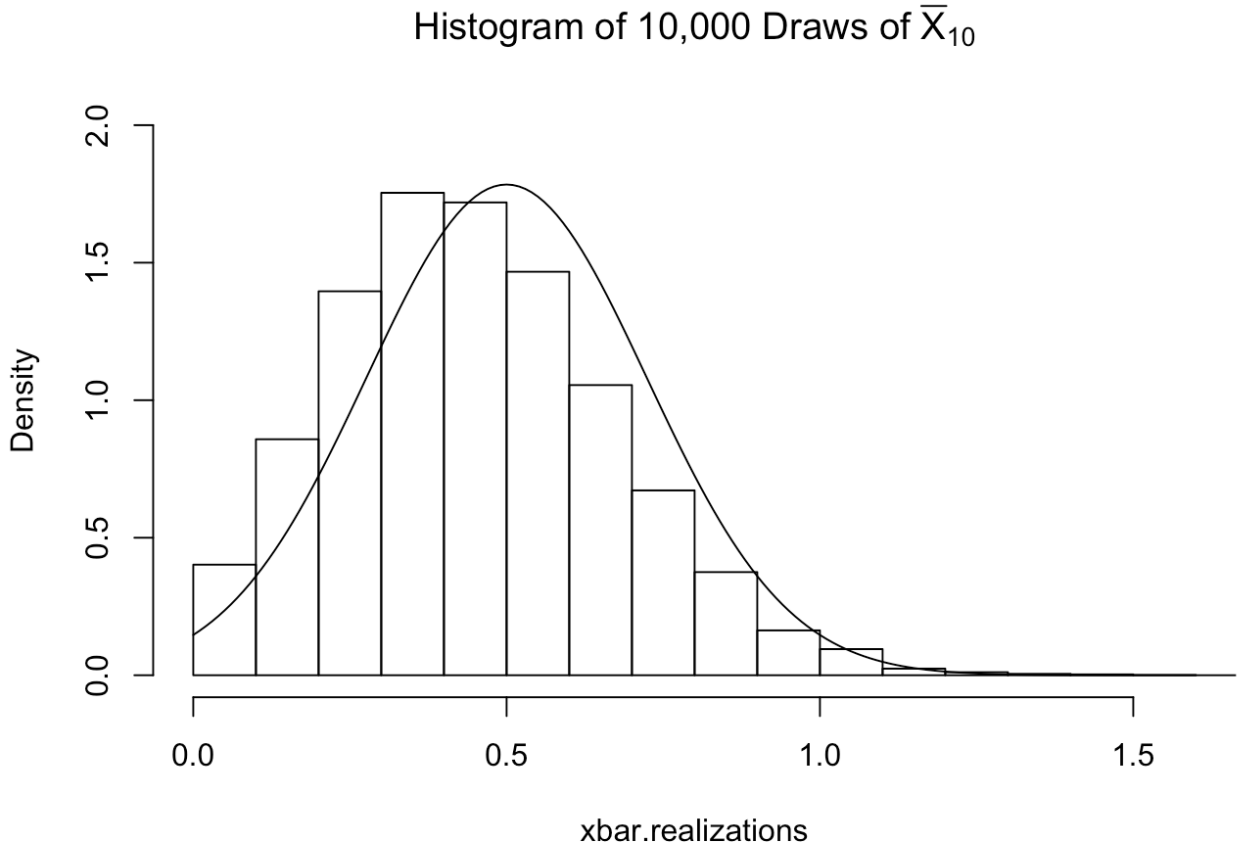
e. Include a code chunk here that defines, `n=3`. In this code chunk, include code that repeats parts (b) - (d) using a random sample of `n` observations instead of 4. Also, as in Problem 4iii), overlay the probability histogram that is created in the last step with the probability density function for a $N(0.5, \sqrt{0.5/n})$ distribution. To do so, use the following:

```
n <-  3
for(i in 1:num.simulations){
  xbar.realizations[i] = mean(rpois(n,0.5))
}

hist(xbar.realizations, freq = F, breaks = 10, main = expression(paste('Histogram
of 10,000 Draws of ',bar(X)[n = 3])))

xvalues = seq(0,2,.01)
yvalues = dnorm(xvalues,.5, sqrt(0.5/n))
lines(xvalues,yvalues)
```

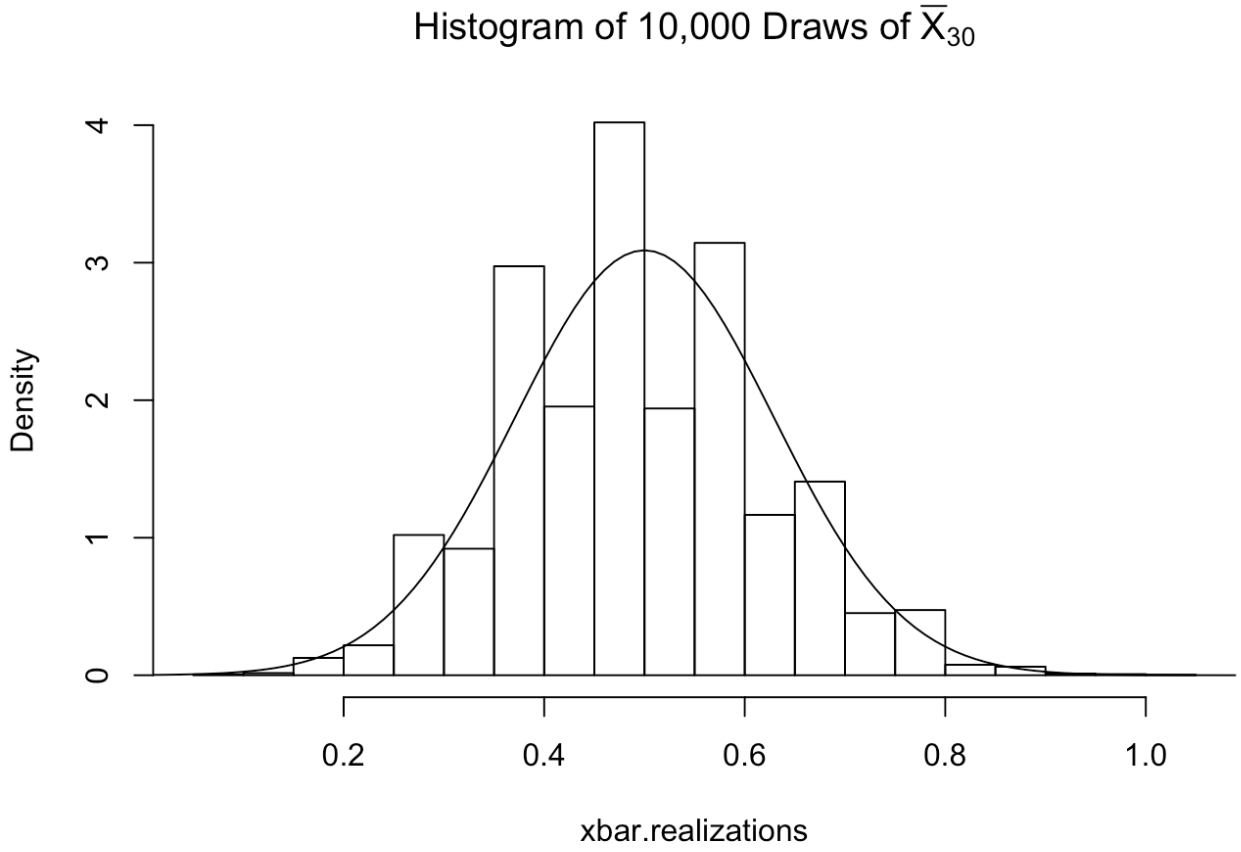


f. Repeat part (e) three times. The first time, set `n=10` . The second time, set `n=30` . The third time, set `n=50` . Be sure you include the probability histogram overlaid with the pdf for $N(0.5, \sqrt{0.5/n})$ for every choice of `n` .

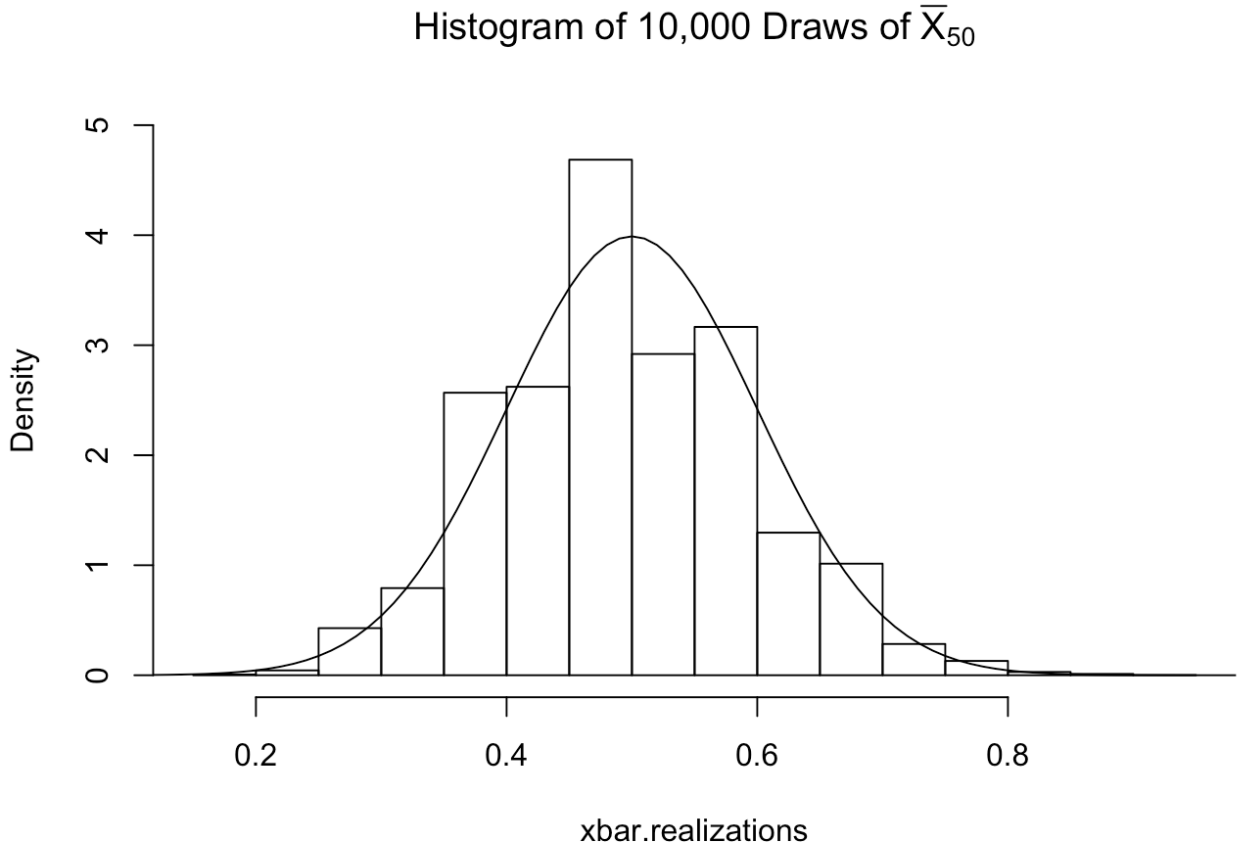
```
n <-  10
for(i in 1:num.simulations){
  xbar.realizations[i] = mean(rpois(n,0.5))
}

hist(xbar.realizations, freq = F, main = expression(paste('Histogram of 10,000 Dr
aws of ',bar(X)[n = 10])), ylim = c(0,2))

xvalues = seq(0,2,.01)
yvalues = dnorm(xvalues,.5, sqrt(0.5/n))
lines(xvalues,yvalues)
```




```
n <-  30
for(i in 1:num.simulations){
  xbar.realizations[i] = mean(rpois(n,0.5))
}

hist(xbar.realizations, freq = F, main = expression(paste('Histogram of 10,000 Dr
aws of ',bar(X)[n = 30])), ylim = c(0,4))

xvalues = seq(0,2,.01)
yvalues = dnorm(xvalues,.5, sqrt(0.5/n))
lines(xvalues,yvalues)
```




```
n <-  50
for(i in 1:num.simulations){
  xbar.realizations[i] = mean(rpois(n,0.5))
}

hist(xbar.realizations, freq = F, main = expression(paste('Histogram of 10,000 Dr
aws of ',bar(X)[n = 50])), ylim = c(0,5))

xvalues = seq(0,2,.01)
yvalues = dnorm(xvalues,.5, sqrt(0.5/n))
lines(xvalues,yvalues)
```


Histogram of 10,000 Draws of $\overline{X}_{50}$
Density
0
1
2
3
4
5
0.2
0.4
0.6
0.8
xbar.realizations