DSCI/MATH 530 RLab Three

Overview

This lab will reinforce the concept of a sampling distribution. Here we will Monte Carlo techniques to verify the sampling distributions in the text and also try some other examples where the theory breaks down.

The key concept here is that we think of a data set as being a sample from a population where the values in the population are described by a probability distribution. That is why this course start out with a prob review.

Submission

Please submit through gradescope. Complete this assignment using R Markdown and converting to pdf. *Make sure each problem is on a separate page.* Use `\newpage` before each problem to insert a page break.

You can also refer to the .Rmd file for this assignment to have an example of this markdown document. Note however, that this version is setup to also use Latex for some math and conversion directly to pdf.

About the term *mean*.

Coming across this term it may be confusing because it has more than one interpretation. Here is a famous example used to illustrate problems of understanding natural language and how a words meaning is important based on its context:

The people went to the *bank*

- to get their money.
- to fish.

bank means two different things depending on how the sentence ends.

mean in statistics is used to refer to the *average* for a sample (i.e. the average in a data set) or as the *expected value* for a distribution. Unfortunately standard deviation is abused the same way. To keep these straight think carefully about whether you are dealing with the data, a finite sample, or a distribution of values, population.

1 Exponential distribution

A common distribution used to positive measurement is the exponential with CDF

$$F(x) = P(X \leq x) = 1 - e^{-\lambda x}$$

This is a good model for the time untill failure of a component or the distribution of rainfall at some locations. It is controlled by just a single parameter, λ and has a mean $1/\lambda$ and variance $1/\lambda^2$. It is easy to show the median, m, of an exponential is

$$1 - e^{-m\lambda} = .5$$

or

$$m = -ln(.5)/\lambda \approx .693/\lambda.$$

$1/\lambda$ is termed a *scale* parameter and if X is an exponential (standard) random variabLle with $\lambda = 1$ then $Y = X/\lambda$ will be exponential with parameter λ. Note that in R λ is refered to as the `rate` argument. Given this nice relationship one wonders why not use e.g. $\theta = (1/\lambda)$ as the parameter of the distribution? We are hostage to the textbook convention!

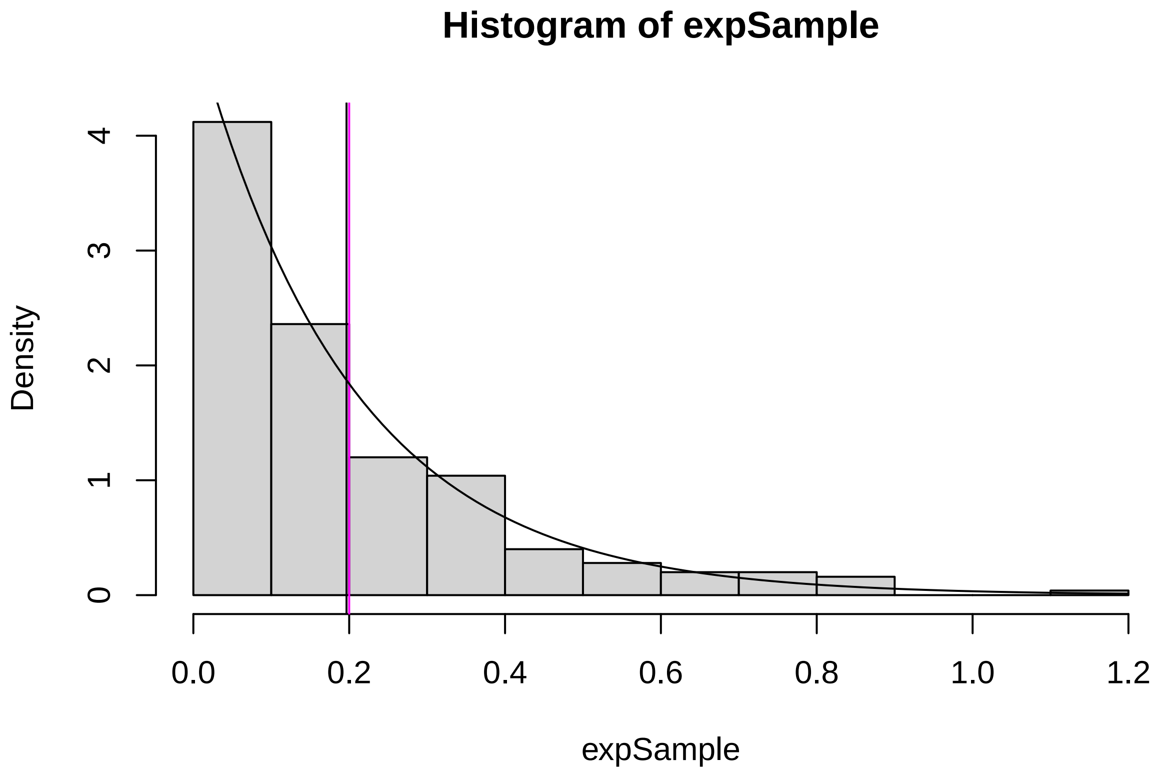
Here is an example of generating a sample of 250 exponential random variables with $\lambda = 5$.

```
set.seed( 232)
expSample<- rexp( 250, rate=5)
```

1(a)

Make a histogram of this sample (using `probability=TRUE`) and add the theoretical exponential density function on top of this. Also add the sample average and the expected value as vertical lines and perhaps with a different color E.g. `abline(v= 1/5, col="magenta")` will add a vertical magenta line at 1/5. *See Rlab1 Question 5 for more details of how to code this up.*

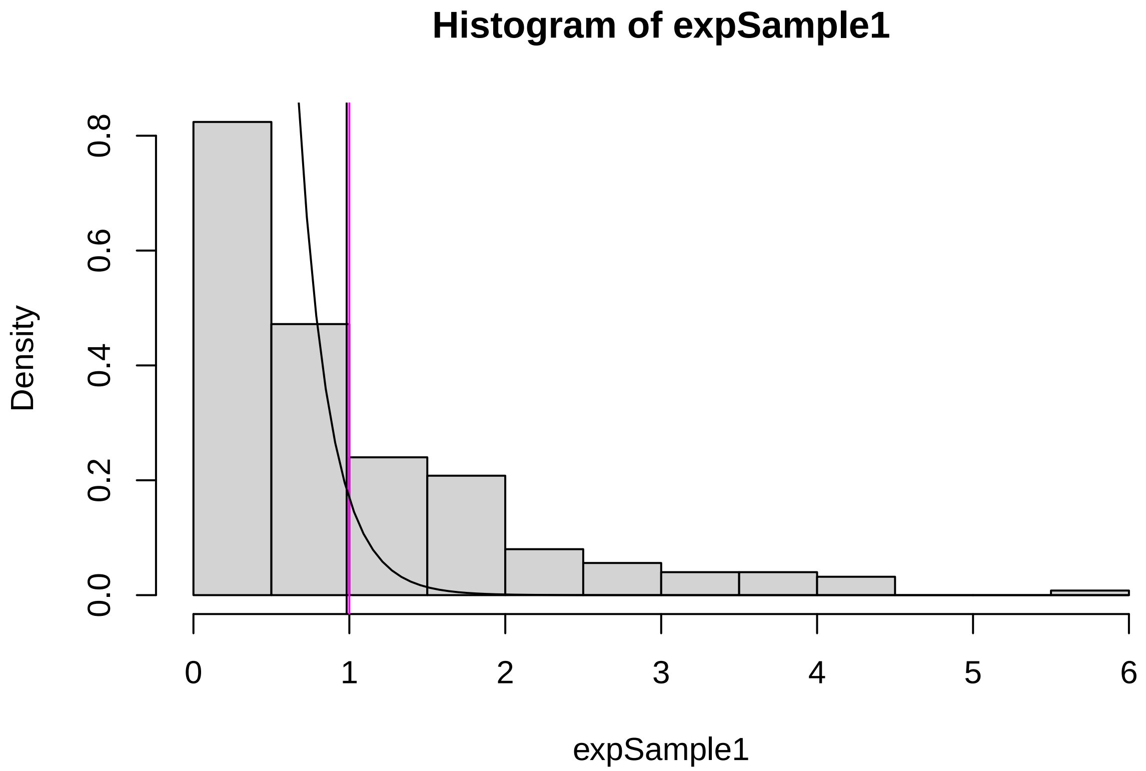
```
h=hist(expSample,probability = TRUE)
abline(v= 1/5, col="magenta")
abline(v=mean(expSample,col="red"))
xlines <-seq(min(h$breaks),max(h$breaks),length.out=100)
lines(x = xlines,y=dexp(xlines,rate=5))
```



1(b)

Repeat (a) but now use the standardized sample `expSample1<- expSample1 * 5`. and add the new sample average and expected value. Does the histogram shape change?

```
expSample1<- expSample * 5
h=hist(expSample1,probability = TRUE)
abline(v= 1, col="magenta")
abline(v=mean(expSample1,col="red"))
xlines <-seq(min(h$breaks),max(h$breaks),length.out=100)
lines(x = xlines,y=dexp(xlines,rate=5)*5)
```


Histogram of expSample1
Density
0.0
0.2
0.4
0.6
0.8
0
1
2
3
4
5
6
expSample1

# 2 Central limit theorem in action.

The main idea behind the central limit theorem is that statistics such as the sample average, and even the sample median, tend to have a regular distribution even if the sample itself is not normally distributed. We will check this using a modest Monte Carlo experiment. The code below generates 500 samples, each is 30 observations from an exponential distribution ($\lambda = 5$). In each case the sample average is found and saved.

```r
M<- 500
n<- 30
set.seed(530)
sampleAverage<- rep( NA, M)
for (k in 1:M ){
  # generate the kth sample of size n
  Y<- rexp( n, rate=5)
  sampleAverage[k]<- mean( Y)
}
```

I am setting the seed so the results are consistent when we rerun the code. Also note that I initially fill the array `sampleAverage` with missing values. This is a useful to spot a bug if not all the entries get filled with numbers.

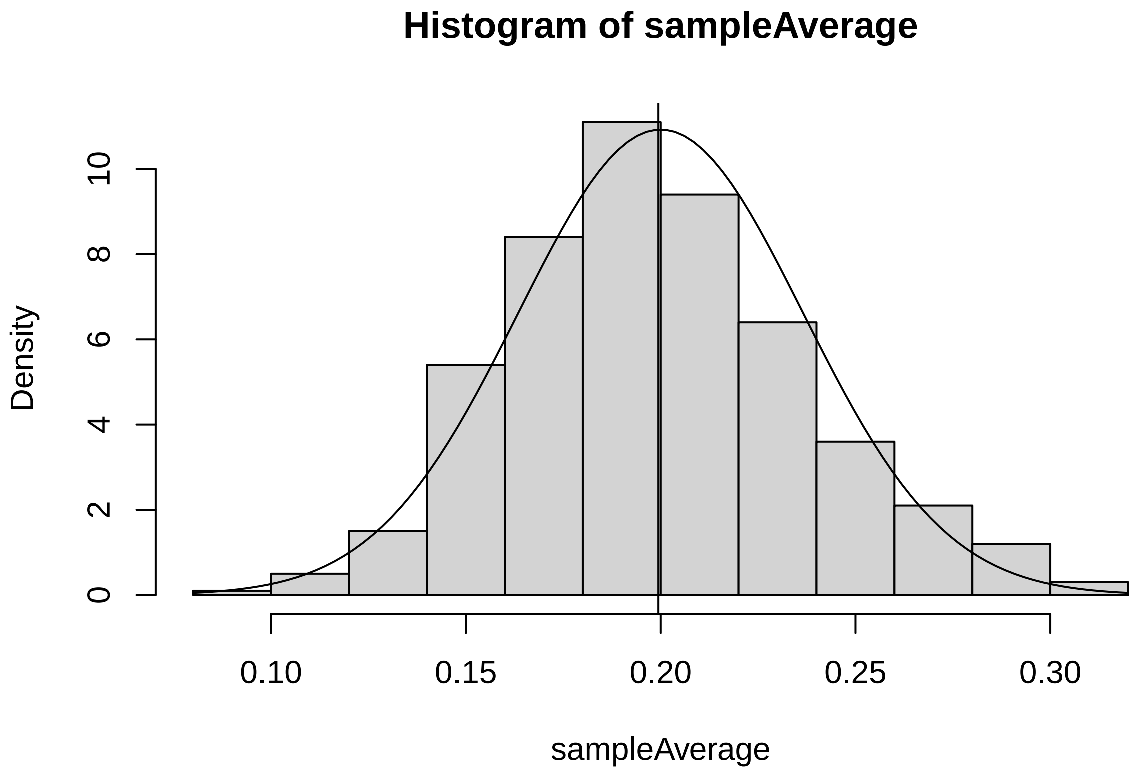
2 (a)

Make a histogram of the sample averages and superimpose a normal density with $\mu = 1/\lambda$ and

$$\sigma = \sqrt{1/n\lambda^2} = 1/(5\sqrt{30})$$

- How well does the normal curve match the histogram?
- Where does the formula for σ come from?
- Compare the `mean(sampleAverage)` with μ and `sd(sampleAverage)` with σ.

```r
h=hist(sampleAverage,probability = TRUE)
abline(v=mean(sampleAverage,col="red"))
xlines <-seq(min(h$breaks),max(h$breaks),length.out=100)
sigma=1/(5*sqrt(30))
mu=1/5
lines(x = xlines,y=dnorm(xlines,mu,sigma))
```



```
sd(sampleAverage)
```

```
## [1] 0.03787992
```

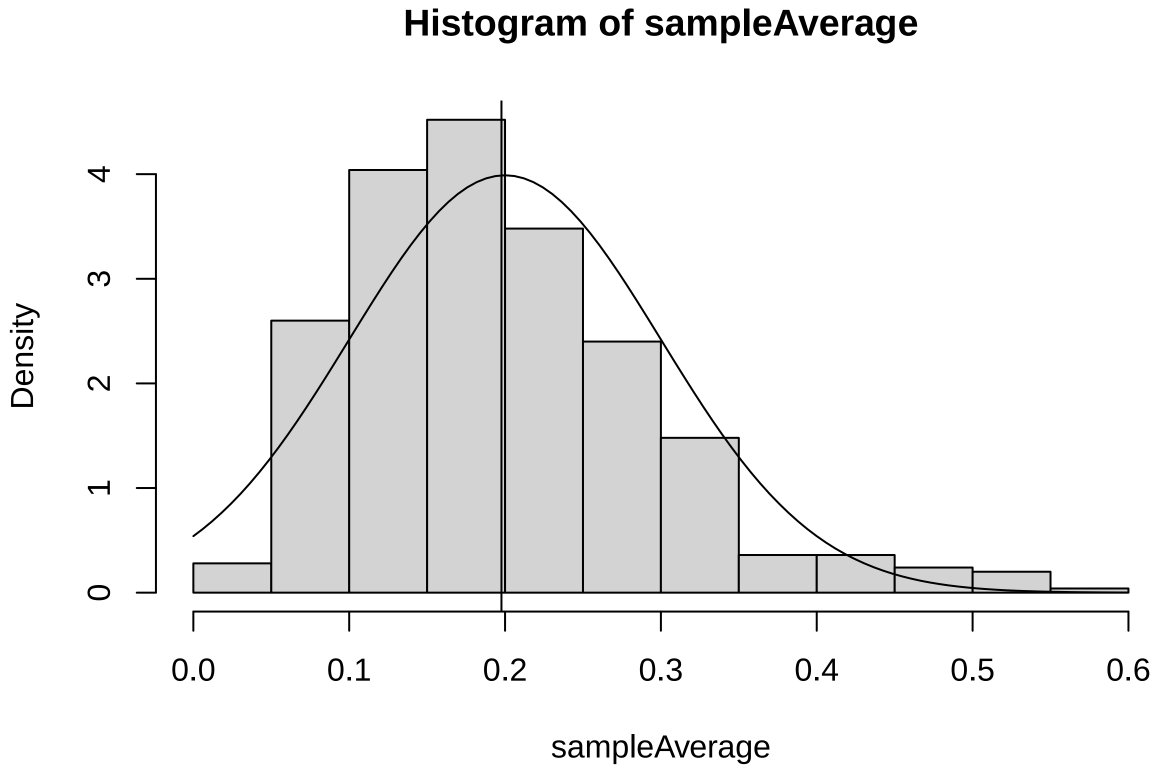
```
mean(sampleAverage)
```

```
## [1] 0.1994016
```

The normal curve matches the histogram well. Both sigmna and muy are similar to the mean and standard deviations of the sample average. Then formula for sigma comes from the sqrt of n over the sd of the sample. Whikch is from the central limit theroem. ## 2 (b) Repeat part (a) except now set `n<- 4` i.e. use a tiny sample size in your Monte Carlo loop but keep `M<-500`. How do your results change?

```
M<- 500
n<- 4
set.seed(530)
sampleAverage<- rep( NA, M)
for (k in 1:M ){
  # generate the kth sample of size n
  Y<- rexp( n, rate=5)
  sampleAverage[k]<- mean( Y)
}
h=hist(sampleAverage,probability = TRUE)
abline(v=mean(sampleAverage,col="red"))
```

```
xlines <-seq(min(h$breaks),max(h$breaks),length.out=100)
sigma=1/(5*sqrt(4))
mu=1/5
lines(x = xlines,y=dnorm(xlines,mu,sigma))
```



```
sd(sampleAverage)
```

```
## [1] 0.09550124
```

```
mean(sampleAverage)
```

```
## [1] 0.1977021
```

This changes sigma and makes the normal curve not fit the histogram as well. This changes sigma to:

```
sigma
```

```
## [1] 0.1
```

3 Breaking the CLT.

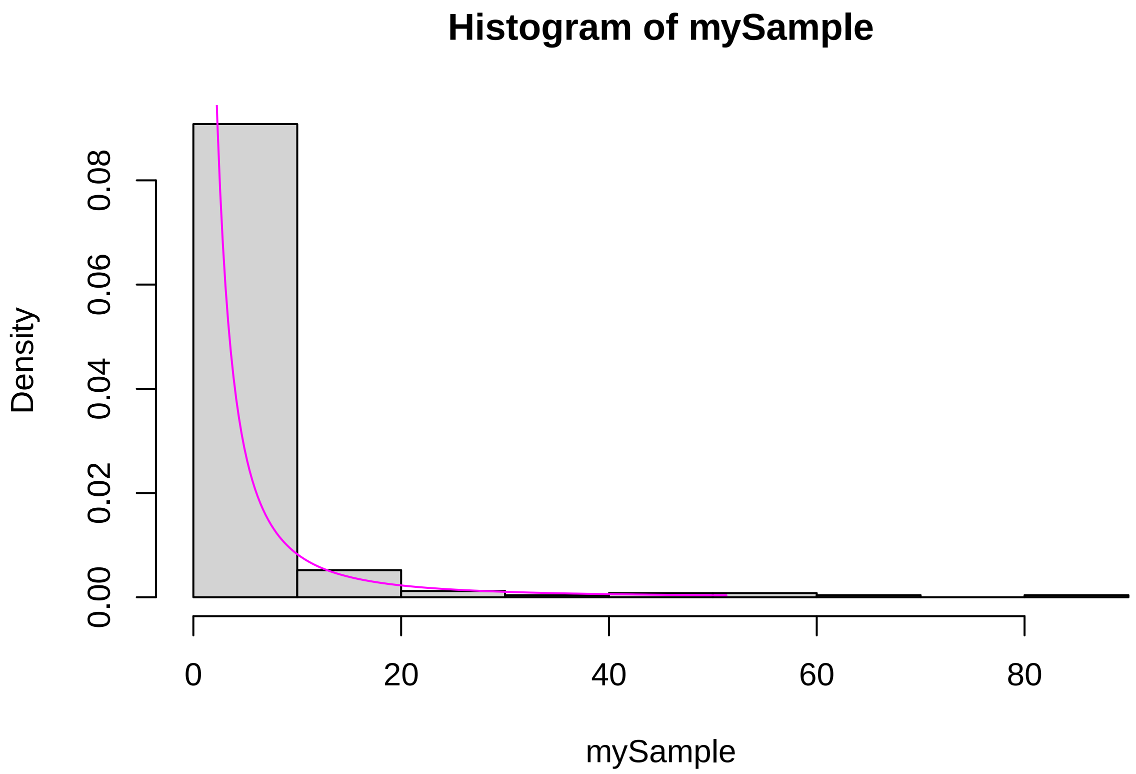
Recall the distribution from exam 1 with probability density function.

$$f(x) = \begin{cases} 0 & x < 0 \\ \frac{1}{(1+x)^2} & x \geq 0 \end{cases}$$

One can show that this distribution has an expected value and a variance that are infinite. But a median (aka 50% quantile) that is 1.0 !. This suggests the average is not a good statistic but maybe the median is.

You can generate a sample of size `n` from this distribution using the example code below.

```r
set.seed(444)
n<- 250
#### simulating sample
U<-  runif(n)
mySample<- U/(1-U)
####
# and take a look at the histogram and density function
# as a sanity check
hist( mySample, probability =TRUE)
grid<- seq( 0, quantile(mySample,.99), length.out=200)
pdf<-  1/ ( 1 + grid)^2
lines( grid, pdf, col="magenta")
```



Wow,skewed or what! This could be a good model for a process that has some rare but very large events. It is similar to the Generalized Pareto family of distributions.

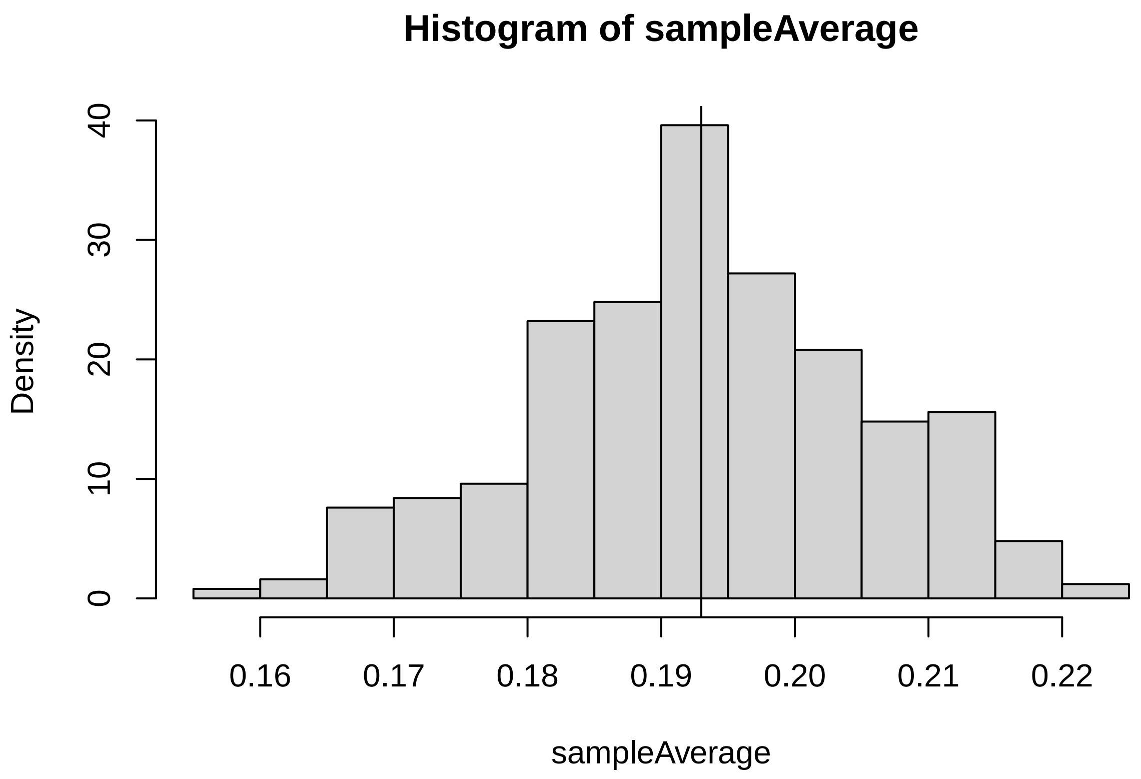
3

Reuse the code from problem 2 but change to this distribution and examine the distribution sample average. Keep `n<- 30` and `M<- 500` Also save the sample median for each sample in another array.

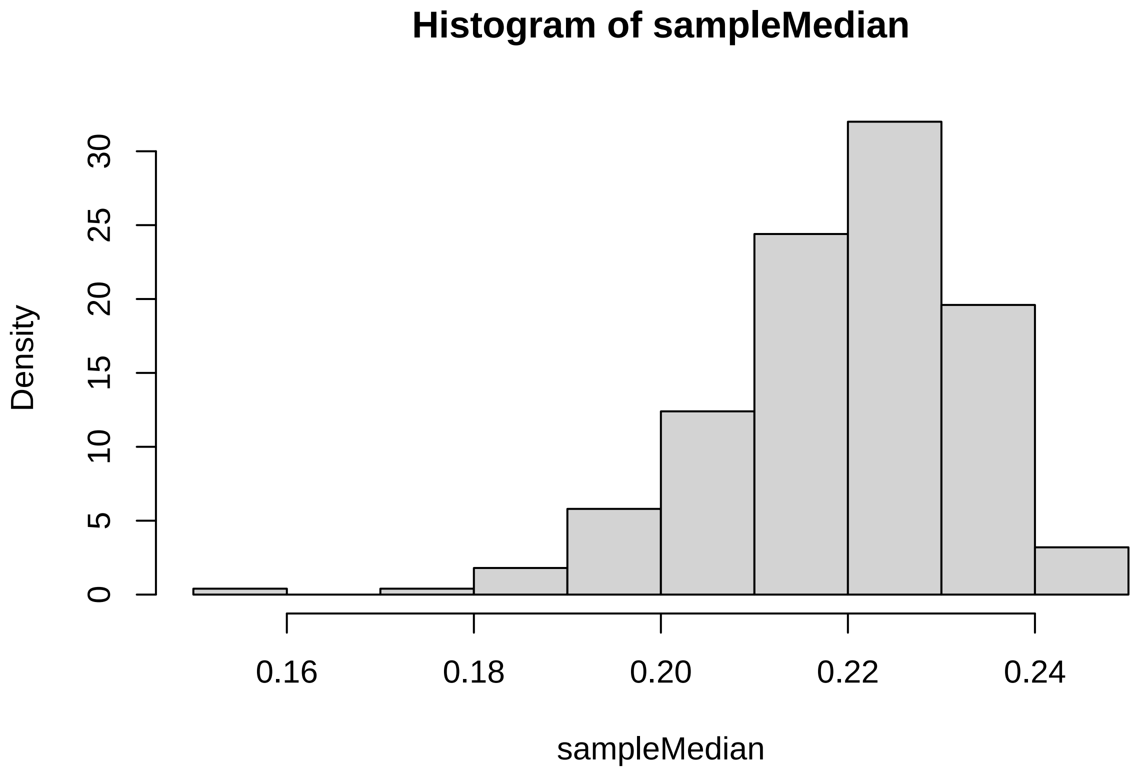
- Does the sample average seem to follow a normal distribution?
- Does the sample median seem to follow a normal distribution? (Hint: Sort of!)

```
M<- 500
n<- 30
set.seed(530)
sampleAverage<- rep( NA, M)
sampleMedian<- rep( NA, M)
pdf <- function(u) u/(1+u)^2

for (k in 1:M ){
  # generate the kth sample of size n
  U<-  runif(n)
  Y <- pdf(U)
  sampleAverage[k]<- mean( Y)
  sampleMedian[k] <- median(Y)
}
h=hist(sampleAverage,probability = TRUE)
abline(v=mean(sampleAverage,col="red"))
```



```
xlines <-seq(min(h$breaks),max(h$breaks),length.out=100)
h=hist(sampleMedian,probability=TRUE)
```



```
sd(sampleAverage)
```

```
## [1] 0.0125162
```

```
mean(sampleAverage)
```

```
## [1] 0.1930031
```

The smaple median is more scewed than th esample average. The sample averagge is pretty normal.

EXTRA CREDIT

Look up the method of simulation using the "inverse CDF" method and explain why this works in Problem 3.

```
num_samples <-  500
U           <-  runif(num_samples)
X           <- U/(1+U)^2
```

This works since sampling from the inverse of the cdf provides information about the pdf.