Homework 1

Zehua Wu

2018/9/28

1. The Birthday Problem

Suppose there are n = 25 people in a room. Assume the following:

- **Ignore leap years and assume there are only 365 days in a year.**
- **Births are uniformly distributed throughout the year.**
- **The people in the room are randomly distributed throughout the year.**

i. What is the probability that two or more of them have the same birthday? Solve this analytically.

$P(same) = 1 - P(different) = 1 - \frac{364}{365} \times \frac{363}{365} \times ... \times \frac{341}{365} \approx 0.5687$

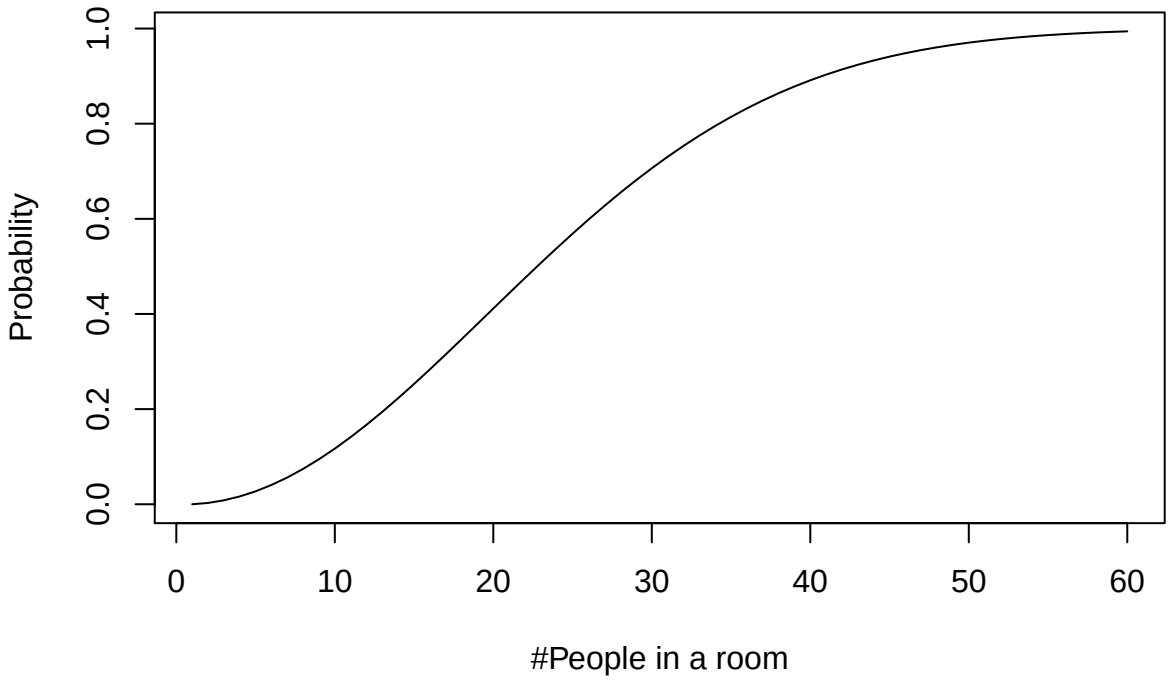
The probability that at least two people share the same birthday when there are 25 people in the room is 56.87%.

ii. Solve [i] computationally in R using the prod function. Next, to better understand the relationship between n and the probability p, plot p against n by looping through rooms of sizes 1 to 60.

```
x=seq(from=341/365, to=364/365, by=1/365)
p=1-prod(x)
print(p)
```

```
## [1] 0.5686997
```

```
n=1
P=vector(mode="numeric")
N=vector(mode="numeric")
for(n in 1:60){
  d=(366-n)/365
  x=seq(from=d, to=1, by=1/365)
  p=1-prod(x)
  P=append(P, p)
  N=append(N, n)
  n=n+1
}
Prob<-data.frame(N, P)
plot(x=N, y=P, xlab="#People in a room", ylab="Probability", type="l")
```



iii. Based on your results from [ii], what is minimum number of people in a room such that the probability of a match is greater than or equal to 50%?

```
print(Prob[Prob$P>=0.5,])
```

```
##     N         P
## 23 23 0.5072972
## 24 24 0.5383443
## 25 25 0.5686997
## 26 26 0.5982408
## 27 27 0.6268593
## 28 28 0.6544615
## 29 29 0.6809685
## 30 30 0.7063162
## 31 31 0.7304546
## 32 32 0.7533475
## 33 33 0.7749719
## 34 34 0.7953169
## 35 35 0.8143832
## 36 36 0.8321821
## 37 37 0.8487340
## 38 38 0.8640678
## 39 39 0.8782197
## 40 40 0.8912318
## 41 41 0.9031516
## 42 42 0.9140305
## 43 43 0.9239229
## 44 44 0.9328854
## 45 45 0.9409759
## 46 46 0.9482528
## 47 47 0.9547744
## 48 48 0.9605980
## 49 49 0.9657796
```

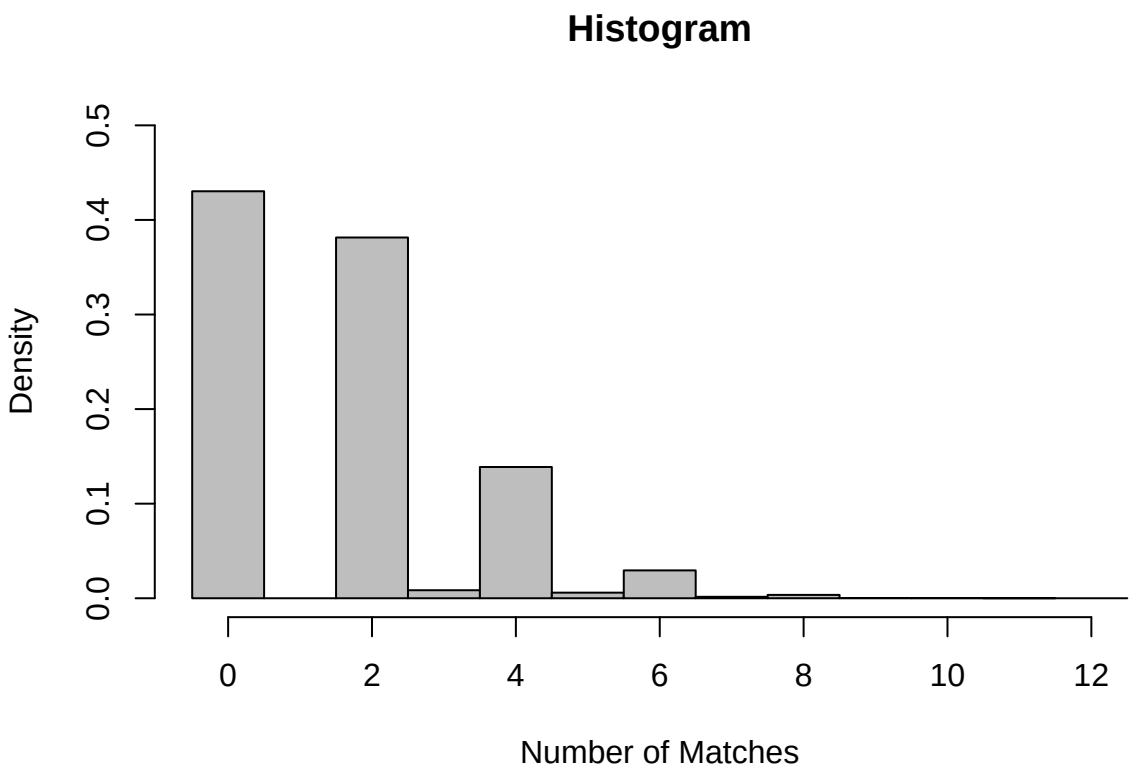
```
## 50 50 0.9703736
## 51 51 0.9744320
## 52 52 0.9780045
## 53 53 0.9811381
## 54 54 0.9838770
## 55 55 0.9862623
## 56 56 0.9883324
## 57 57 0.9901225
## 58 58 0.9916650
## 59 59 0.9929894
## 60 60 0.9941227
```

The minimum number of people needed to be in the room for the probability to be at least 50% is 23.

2. The Birthday Problem -Again

Based on the same assumptions of Problem 9, write a simulation in R that generates 100,000 simulated rooms of 25 people, and plot a histogram of the density vs. the number of birthday matches. This histogram represents your approximation of the distribution of birthday matches. Do your results agree with Problem 9? *Hint: Consider using the R functions sample and unique to make the calculations easier.*

```
rm(list=ls())
birthday<-c(1:365)
M<-vector()
for(i in 1:100000){
  room<-sample(birthday, size=25, replace=TRUE)
  M[i]<-length(subset(room, duplicated(room)|duplicated(room, fromLast=T)))
}
cut<-(0:(max(M)+1))-0.5
hist(M, xlab="Number of Matches", main="Histogram", freq=F, breaks=cut, col=8, ylim=c(0.0, 0.5))
```



3. R Exercise:

Plot a histogram of monthly returns of your favorite stock (use at least 10 years of data), and fit an appropriate distribution. Comment on your fit. Use this distribution to compute P(r > 10%), where r =return.

```
rm(list=ls())
library(quantmod)
library(tseries)
library(stats)
library(MASS)
library(sfsmisc)
library(distrEx)
library(fitdistrplus)
```

```
AMZN<-get.hist.quote("AMZN", compression="m", retclass="zoo", quote="AdjClose")
```

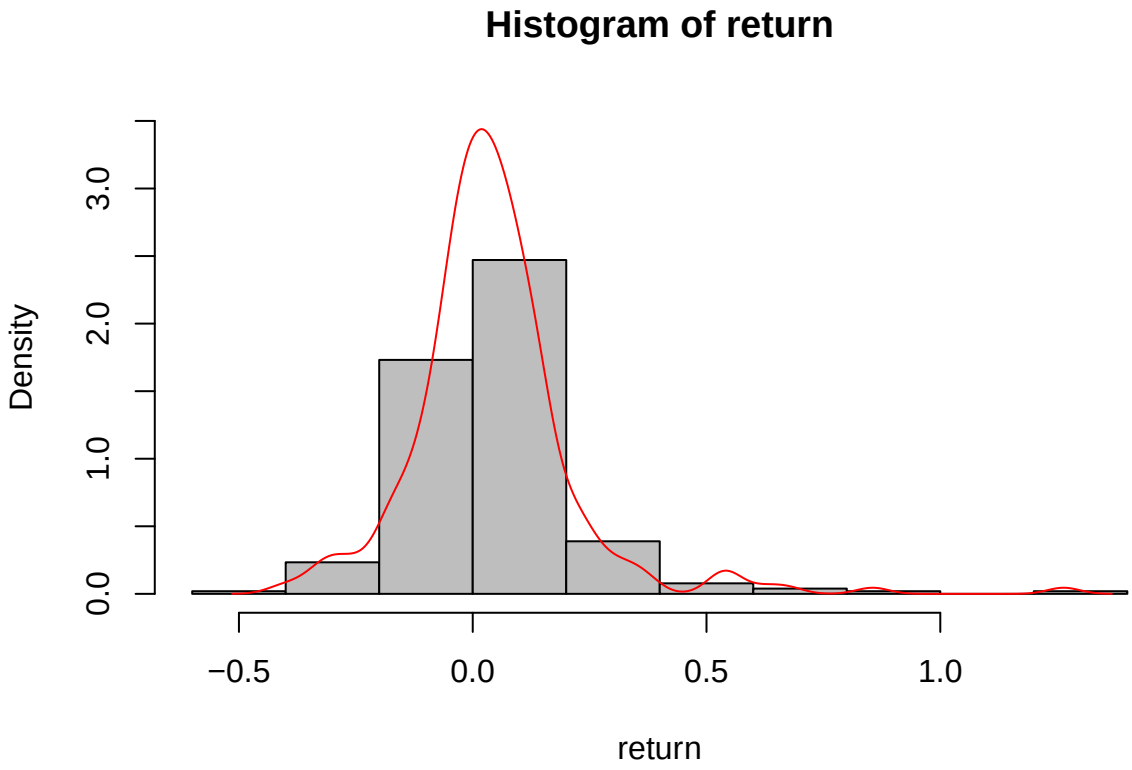
```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
##
## WARNING: There have been significant changes to Yahoo Finance data.
## Please see the Warning section of '?getSymbols.yahoo' for details.
##
```

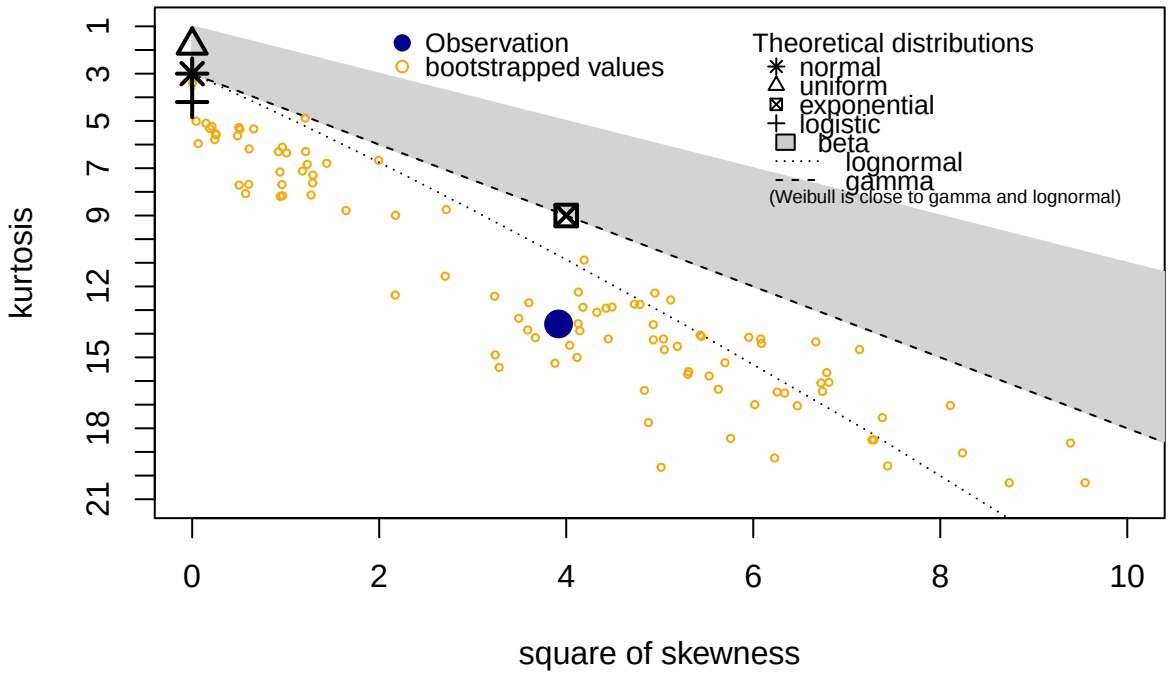
```
## This message is shown once per session and may be disabled by setting
## options("getSymbols.yahoo.warning"=FALSE).

## time series starts 1997-05-01
## time series ends   2018-10-01
```

```
n<-length(AMZN$Adjusted)
post<-as.numeric(AMZN$Adjusted[2:n])
prior<-as.numeric(AMZN$Adjusted[1:n-1])
return<-(post-prior)/prior
```

```
hist(return, col=8, freq=F, ylim=c(0,3.5))
lines(density(return), col="red")
```




```
descdist(return, boot=100)
```




```
## summary statistics
## ------
## min:  -0.4115523   max:  1.26383
## median:  0.02618157
## mean:  0.04193082
## estimated sd:  0.1788532
## estimated skewness:  1.979414
## estimated kurtosis:  13.58595
```

Comments: The Cullen and Frey graph tells us that the distribution our stock return data follows is pretty close to a lognormal distribution. However, since lognormal distributions should be nonzero, we can transform our data such that it no longer contains values smaller than 0. We can do this by adding 0.5 to each value in returns data and run a Kolmorov- test again.

```
return1<-return+0.5
y<-fitdistr(return1, densfun="lognormal")
fit<-rlnorm(n=length(y), meanlog=y[["estimate"]][["meanlog"]], sdlog=y[["estimate"]][["sdlog"]])
ks.test(return1, fit)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  return1 and fit
## D = 0.32607, p-value = 0.5723
## alternative hypothesis: two-sided
```

Comments: Since our p-value is rather large, we fail to reject the null hypothesis that return follos a log normal distribution. We can then calculate $P(r > 0.1)$ using lognormal distribution. But note that since lognormal distribution is nonnegative, we should take this into consideration when calculating $P(r > 0.1)$ using the plnorm() function.

```r
P<-plnorm(q=0.1+abs(min(return)), meanlog=y[["estimate"]][["meanlog"]], sdlog=y[["estimate"]][["sdlog"]]
cat("The probability that return is higher than 10% is", P, "\n")
```

```
## The probability that return is higher than 10% is 0.5067238
```

4.

Let X and Y have the joint density function $f_{X,Y}(x,y) = cx(y-x)e^{-y}$, $0 \le x \le y < \infty$

(a)Find c.

Set $\int_0^\infty \int_x^\infty f_{X,Y}(x,y)\, dy\, dx = 0$

We get $c = 1$

So $f_{X,Y}(x,y) = x(y-x)e^{-y}$

(b)Find $f_{X|Y}(x|y)$ and $f_{Y|X}(y|x)$.

$f_X(x) = \int_x^\infty f_{X,Y}(x,y)\, dy = xe^{-x}$

$f_Y(y) = \int_0^y f_{X,Y}(x,y)\, dx = \frac{1}{6}y^3e^{-y}$

So $f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)} = \frac{x(y-x)e^{-y}}{\frac{1}{6}y^3e^{-y}} = \frac{6x(y-x)}{y^3}$

$f_{Y|X}(y|x) = \frac{f_{X,Y}(x,y)}{f_X(x)} = \frac{x(y-x)e^{-y}}{xe^{-x}} = \frac{(y-x)e^{-y}}{e^{-x}}$

(C)Find $E(X|Y)$ and $E(Y|X)$.

$E(X|Y) = \int_0^y x \cdot \frac{6x(y-x)}{y^3}\, dx = \frac{y}{2}$

$E(Y|X) = \int_x^\infty y \cdot \frac{(y-x)e^{-y}}{e^{-x}}\, dy = x + 2$

5. Evans & Rosenthal: 4.1.11

Suppose that $X_1, X_2, \ldots, X_{10}$ is an i.i.d. sequence from an $N(0,1)$ distribution. Generate a sample of $N = 10^3$ values from the distribution of $\max(X_1, X_2, \ldots, X_{10})$. Calculate the mean and standard deviation of this sample.

```r
library(prob)
```

```r
rm(list=ls())
X<-matrix(NA, 1000, 10)
M<-vector()
for(i in 1:1000){
  X[i,]<-rnorm(n=10, mean=0, sd=1)
  M[i]<-max(X[i,])
}
cat("The mean of this sample is", mean(M), "\n")
```

```
## The mean of this sample is 1.546962
```

```r
cat("The standard deviation of this sample is", sd(M), "\n")
```

```
## The standard deviation of this sample is 0.57463
```

6. Evans & Rosenthal: 4.2.12

Generate i.i.d. $X_1, \ldots, X_n$ distributed Exponential(5) and compute M_n when $n = 20$. Repeat this N times, where N is large (if possible, take $N = 10^5$, otherwise as large as is feasible), and compute the proportion of values of M_n that lie between 0.19 and 0.21. Repeat this with $n = 50$. What property of convergence in probability do your results illustrate?

```
rm(list=ls())
set.seed(100)
X1<-matrix(NA, 100000, 20)
M1<-vector()
for(i in 1:100000){
  X1[i,]<-rexp(n=20, rate=5)
  M1[i]<-mean(X1[i,])
}

X2<-matrix(NA, 100000, 50)
M2<-vector()
for(i in 1:100000){
  X2[i,]<-rexp(n=50, rate=5)
  M2[i]<-mean(X2[i,])
}

proportion1<-length(M1[M1>=0.19 & M1<=0.21])/length(M1)
proportion2<-length(M2[M2>=0.19 & M2<=0.21])/length(M2)

cat("The propotion of values that lie in between 0.19 and 0.21 when n is 20 is", proportion1, "\n")
```

```
## The propotion of values that lie in between 0.19 and 0.21 when n is 20 is 0.17824
```

```
cat("The propotion of values that lie in between 0.19 and 0.21 when n is 50 is", proportion2, "\n")
```

```
## The propotion of values that lie in between 0.19 and 0.21 when n is 50 is 0.27418
```

Comments: This result shows that when the sample size n is greater, a greater proportion of sample means will fall in the interval [0.19, 0.21], which contains the expected value of Exponential(5). This agrees with convergence in probability.

```
X3<-matrix(NA, 100000, 10000)
M3<-vector()
for(i in 1:100000){
  X3[i,]<-rexp(n=10000, rate=5)
  M3[i]<-mean(X3[i,])
}
proportion3<-length(M3[M3>=0.19 & M3<=0.21])/length(M3)
cat("The propotion of values that lie in between 0.19 and 0.21 when n is 50 is", proportion3, "\n")
```

```
## The propotion of values that lie in between 0.19 and 0.21 when n is 50 is 1
```

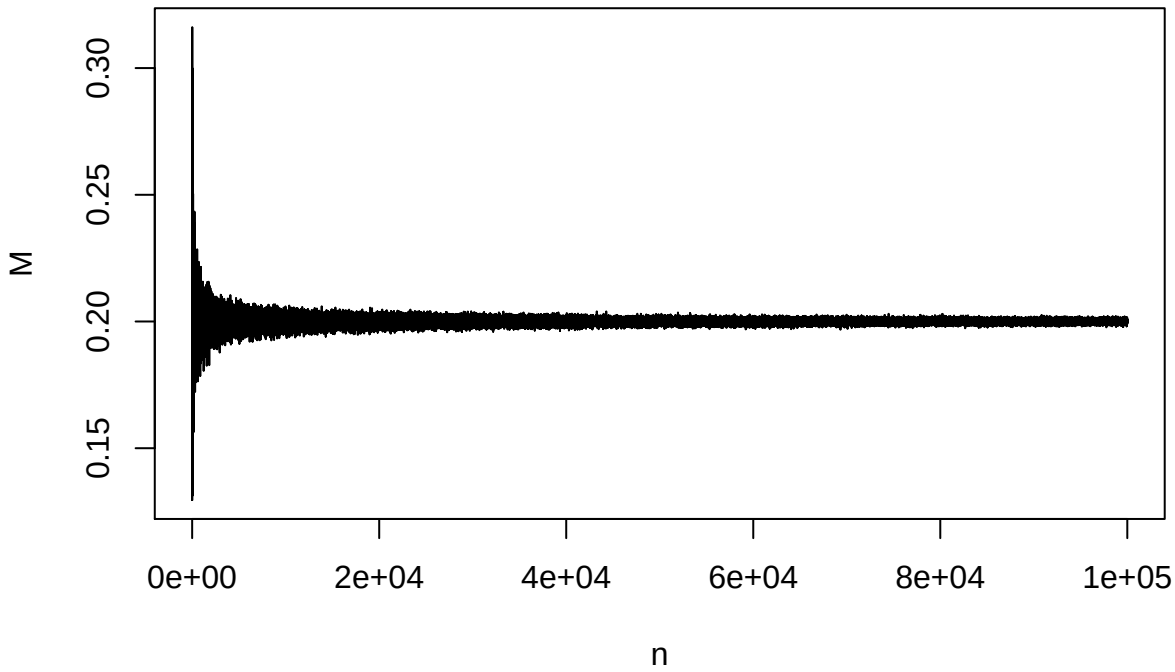
Comments: Further, we can see that when n is extremely large (in this case $n = 10000$), the proportion of values that lie in between 0.19 and 0.21 is almost 1.

7. Evans & Rosenthal: 4.3.13

Generate i.i.d. $X_1, \ldots, X_n$ distributed Exponential(5) with n large (take $n = 10^5$ if possible). Plot the vales $M_1, M_2, \ldots, M_n$. To what value are they converging? How quickly?

```r
rm(list=ls())
M<-vector()
for(i in 1:100000){
  X<-rexp(i, rate=5)
  M[i]<-mean(X)
}
```

```r
plot(M, type='l', xlab='n', ylab='M')
```

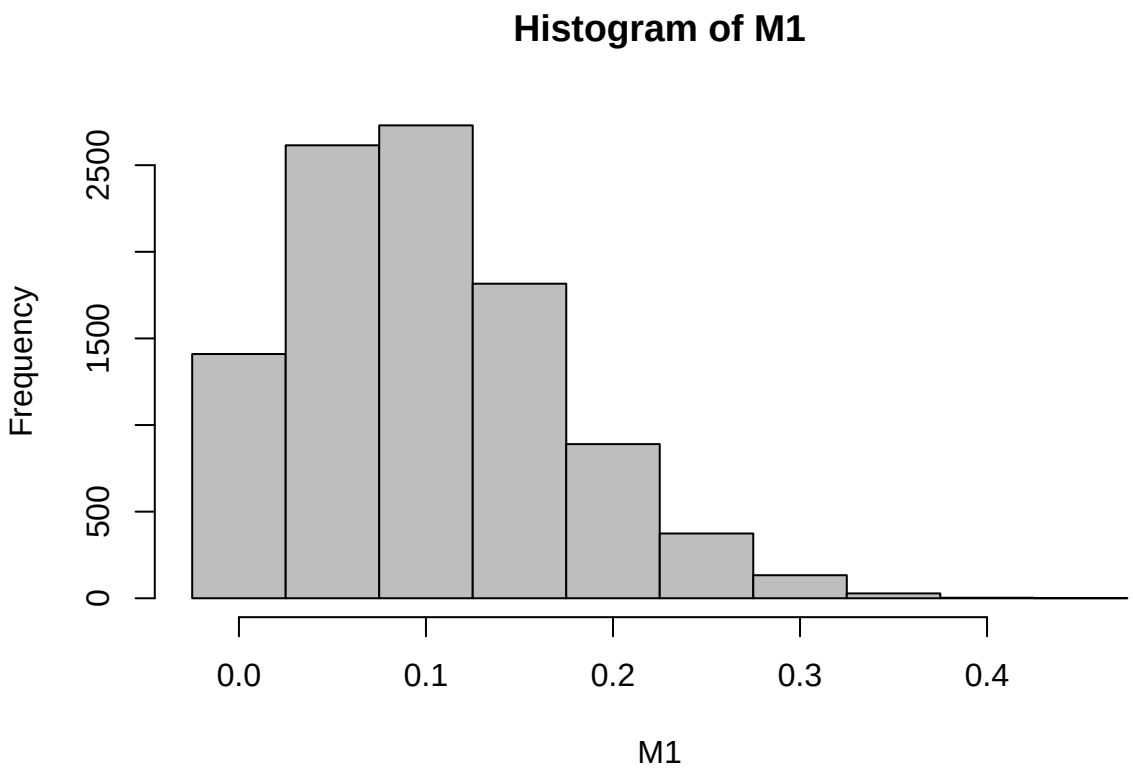


Comment: M_n is converging to the expected value of Exponential(5), which is 0.2.

8. Evans & Rosenthal: 4.4.19

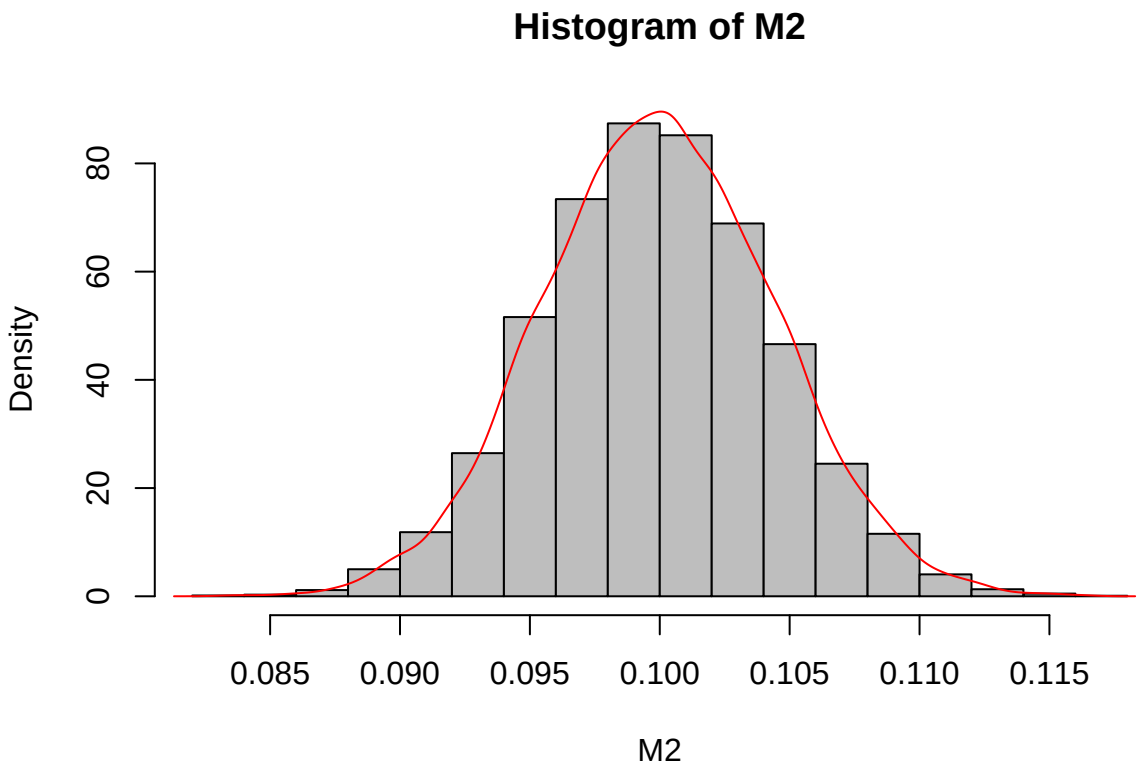
Generate N samples $X_1, X_2, \ldots, X_{20}$ from the Binomial(10, 0.01) distribution for N large ($N = 10^4$, if possible). Use these samples to construct a density histogram of the values of M_{20}. Comment on the shape of this graph.

```r
rm(list=ls())
set.seed(100)
X1<-matrix(NA, 10000, 20)
M1<-vector()
for(i in 1:10000){
  X1[i,]<-rbinom(n=20, size=10, prob=0.01)
  M1[i]<-mean(X1[i,])
}
cut=seq(from=min(M1), to=max(M1)+0.05, by=0.05)-0.025
hist(M1, breaks=cut, col=8)
```



Comments: According to Central Limit Theorem, when n is large, the sum of $X_1 + ... + X_n$ should converge to a normal distribution. Although in the histogram, the left tail is cut off, we can see that the shape looks like a normal distribution. We can further illustrate this by increasing n. For example, we can look at the case when $n = 1000$

```
X2<-matrix(NA, 10000, 5000)
M2<-vector()
for(i in 1:10000){
  X2[i,]<-rbinom(n=5000, size=10, prob=0.01)
  M2[i]<-mean(X2[i,])
}
hist(M2, col=8, freq=F)
lines(density(M2), col="red")
```



Comments: Here we can see that both the histogram and the density plot show a distribution that looks closer to a normal distribution with $\mu = 0.1$. We can test the normality with a Kolmogorov-Smirnov Tests.

```
library(fitdistrplus)
fit<-fitdistr(M2, "normal")
normal<-rnorm(n=length(M2), mean=fit[["estimate"]][["mean"]], sd=fit[["estimate"]][["sd"]])
ks.test(M2, normal)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  M2 and normal
## D = 0.0171, p-value = 0.1074
## alternative hypothesis: two-sided
```

Comments: Since the p-value is rather small, we should reject the null hypothesis and accept that M_{1000} is normally distributed. This agrees with the Central Limit Theorem.