Economics 403A: Homework 1

Ross Lewis

September 28, 2018

1 - i

The probability of two people having different birthdays is

`364/365`

That is to say, the probability of them haveing the same birhtday is

`1 - 364/365`

However, as you increase the number of people in the room, the combinations increase quickly. There will always be

`N choose 2`

pairs, where N is the number of people in the room. So, when N is 25, the probability that two or more people will have the same birthday is

```
1 - (364/365)^choose(N,2)
= 1 - (364/365)^(25!/(2!*23!))
= 1 - (364/365)^300
= .5609078
```

1 - ii

To solve the same problem programatically, we can use the
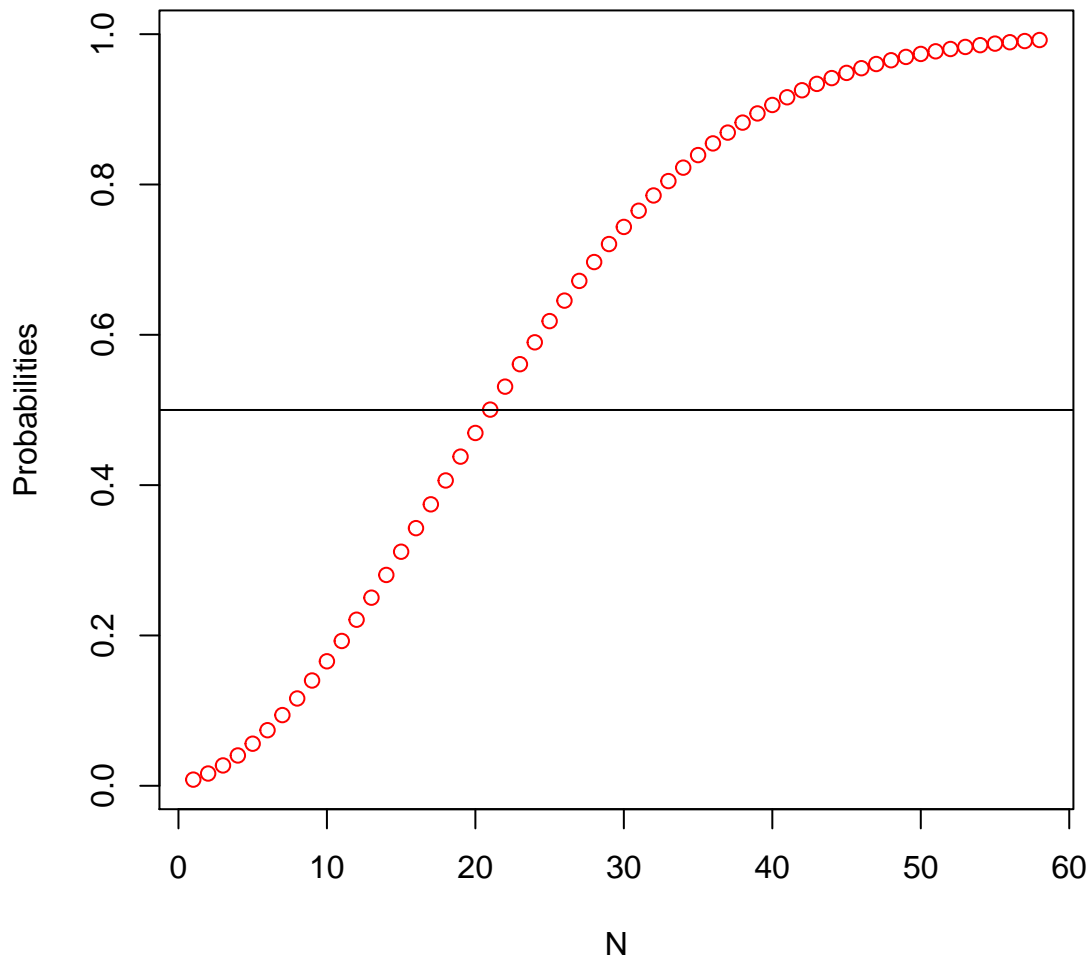
`prod`

function to replace the factorial operations:

`1 - (364/365)^(prod(1:25)/(prod(1:2)*prod(1:23)))`

With this, we get the same answer of 0.5609078. Now that we have an equation for the probability, we can loop through different values of N to find the probabilities of of two people having the same birthday in a room that large:

```
probs = c()
for (i in 3:60){
  prob = 1 - (364/365)^(prod(1:i)/(prod(1:2)*prod(1:(i-2))))
  probs = append(probs,values = prob)
}

plot(probs,
     main='Probability that Two People Will Have the Same Birthday',
     xlab='N',ylab='Probabilities',col='red')
abline(.5,0)
```

Probability that Two People Will Have the Same Birthday

1 - iii

Visually, it looks like the probability of 0.5 is acheived between 20 and 25. We can test this:

```r
for (i in 20:25){
  prob = 1 - (364/365)^(prod(1:i)/(prod(1:2)*prod(1:(i-2))))
  cat("The probability at N = ",i,"is",prob,'\n')
}
```

```
## The probability at N =  20 is 0.4062295
## The probability at N =  21 is 0.4379318
## The probability at N =  22 is 0.4693992
## The probability at N =  23 is 0.5004772
## The probability at N =  24 is 0.5310233
## The probability at N =  25 is 0.5609078
```

With this output, we can see that the minimum number of people needed in a room such that the probability of two of them having the same birthday is greater than 0.5 is 23.

2

Below is the figure showing the distribution of matches for 10,000 simulations of 25 people in a room:

```r
numMatches = c()
for(i in 1:10000){
  sam = sample(1:365,25,replace = T)
  numMatches = append(numMatches,25-length(unique(sam)))
}
hist(numMatches,main="Distribution of Birthday Matches",
     xlab="Number of matches",ylab="Density")
```

Distribution of Birthday Matches
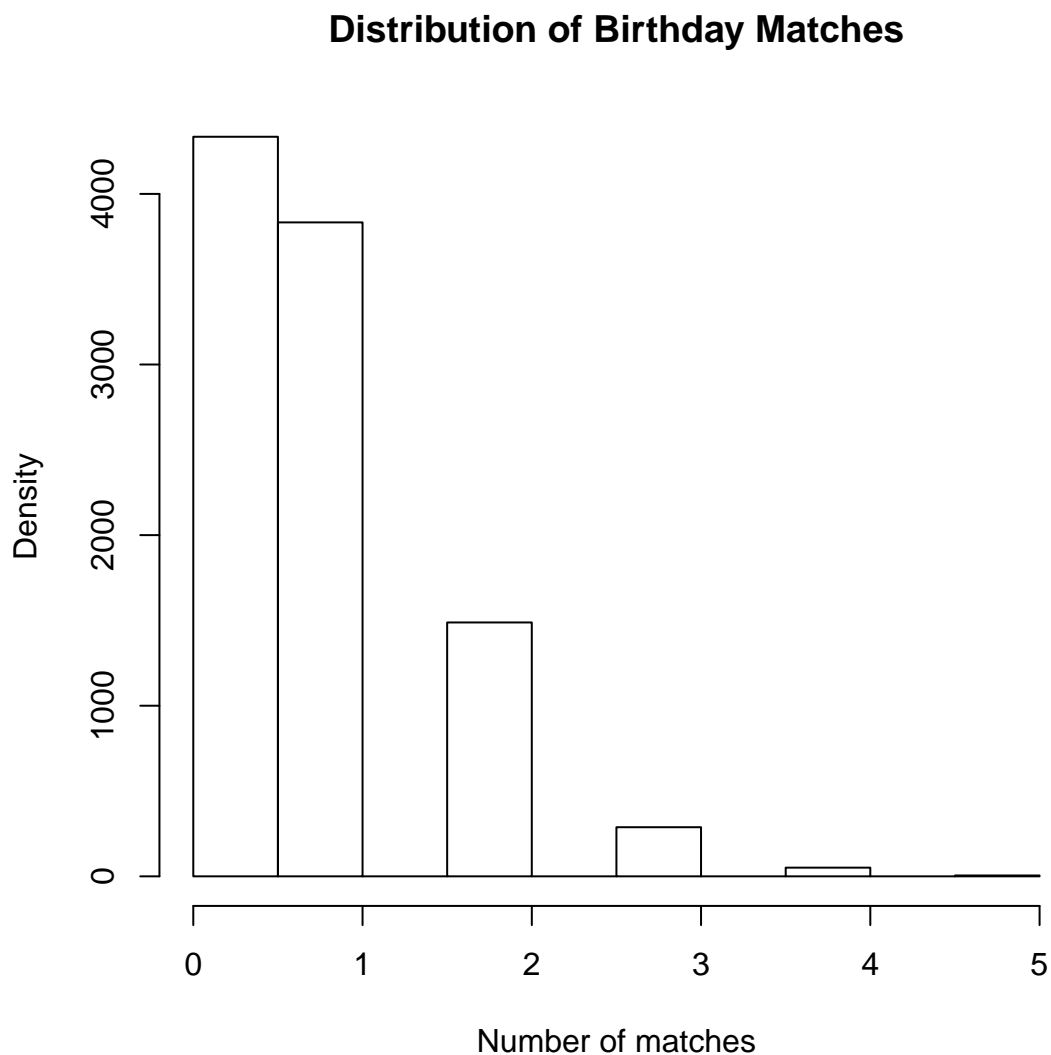
For additional clarity, here is the simulation's resulting percentages of no match vs match rooms:

```r
#Rooms with a match
length(which(numMatches > 0))/10000
```

```
## [1] 0.5665
```

```
#Rooms without a match
length(which(!numMatches > 0))/10000
```

[1] 0.4335

My results are almost exactly as predicted using the analytical method.

3

Here is a histogram of Microsoft's monthly returns from 2004 to 2018 along with a distribution description:

```
library(Quandl)
library(fitdistrplus)
```

```
#Download monthly stock prices and dividends
my_api_key = "sBTb9YZ2pF-6NyfSBSxA"
msftPrice<-Quandl("EOD/MSFT", api_key=my_api_key, collapse="monthly",start_date="2004-12-31")
msftDiv = Quandl("EOD/MSFT", api_key=my_api_key,start_date="2004-12-31")

#add the dividends to the monthly prices to get a total return
for (row in which(msftDiv$Dividend > 0)){
  msftPrice[which(format(as.Date(msftPrice$Date), "%Y-%m") ==
                    format(as.Date(msftDiv[row,]$Date), "%Y-%m")),]$Dividend = msftDiv[row,]$Dividend
}

#calculate the total return and the percentage change
msftPrice$return = c(-diff(msftPrice$Open, lag = 1),NA)
msftPrice = na.omit(msftPrice)
msftPrice$return = msftPrice$return + msftPrice$Dividend
msftPrice$percRet = msftPrice$return/(msftPrice$Open-msftPrice$return)

#check which distribution my data is closest to
descdist(msftPrice$percRet, boot = 1000)
```

Cullen and Frey graph
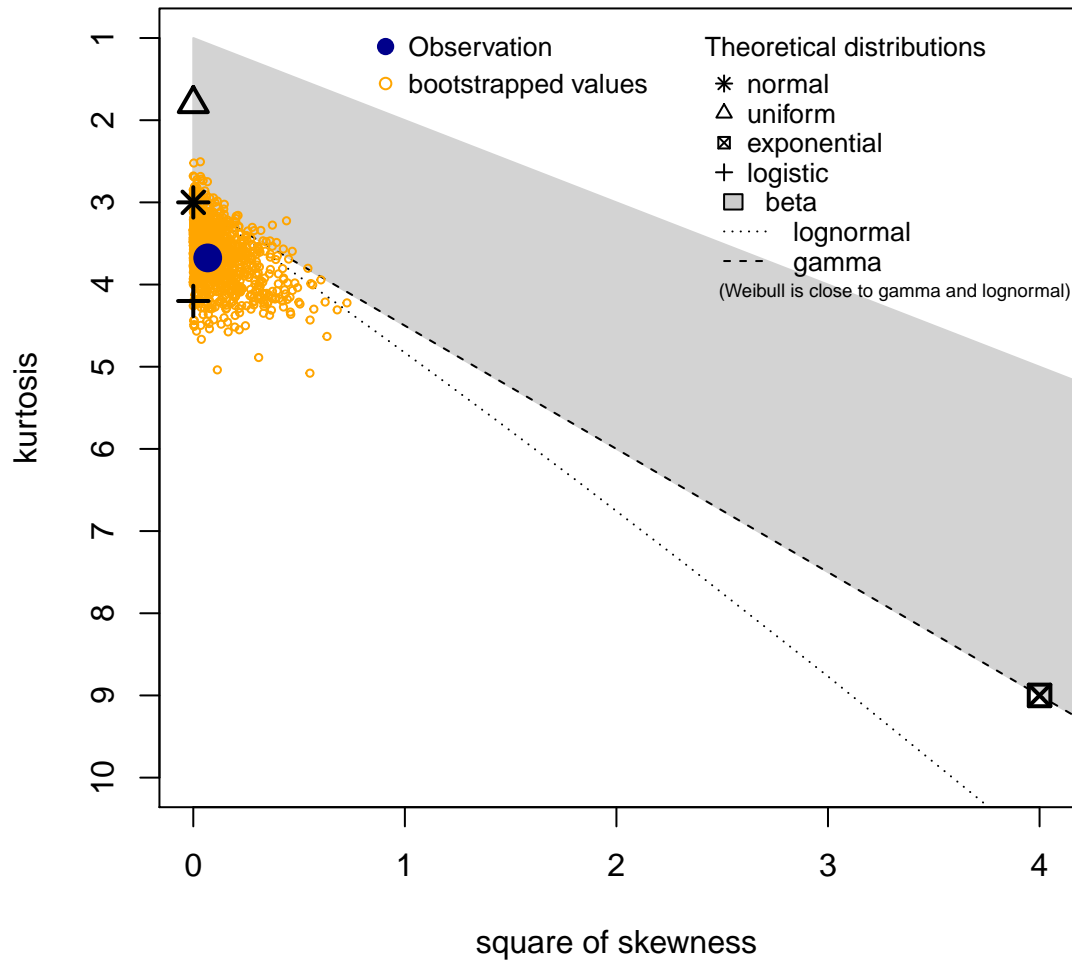

```
## summary statistics
## ------
## min:  -0.1648352   max:  0.2151322
## median:  0.01596686
## mean:  0.01273337
## estimated sd:  0.0649215
## estimated skewness:  0.2611399
## estimated kurtosis:  3.675947
```

```r
#create the historgram
par(mfrow = c(2, 1))
hist(msftPrice$percRet,main="Microsoft's Monthly Returns",
     xlab="Returns",col=c('blue','red','yellow','green'),breaks = 20)

#My data was close to normal, logistic, lognormal, and gamma distributions.
#However, I have negative values so I fit to a normal.
norm <- fitdist(msftPrice$percRet, "norm")
print(norm)
```
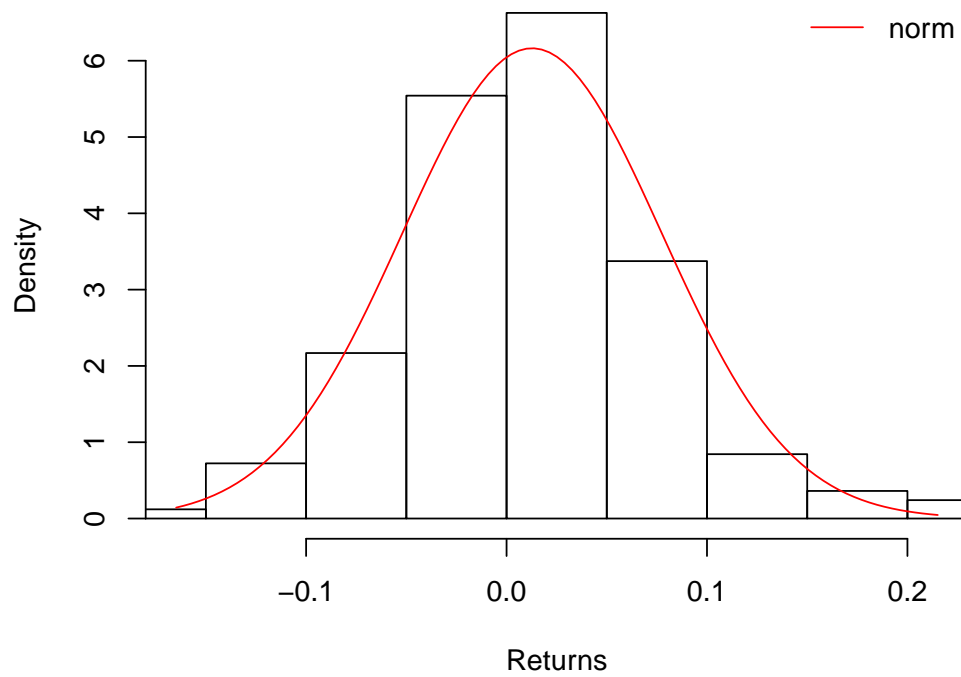
```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters:
##         estimate  Std. Error
## mean 0.01273337 0.005023685
## sd   0.06472566 0.003548467
```

```r
plot.legend <- c("norm")
denscomp(list(norm), legendtext = plot.legend,
         main='Normal Distribution Fit to Microsoft Returns',
         xlab='Returns')
```

Microsoft's Monthly Returns
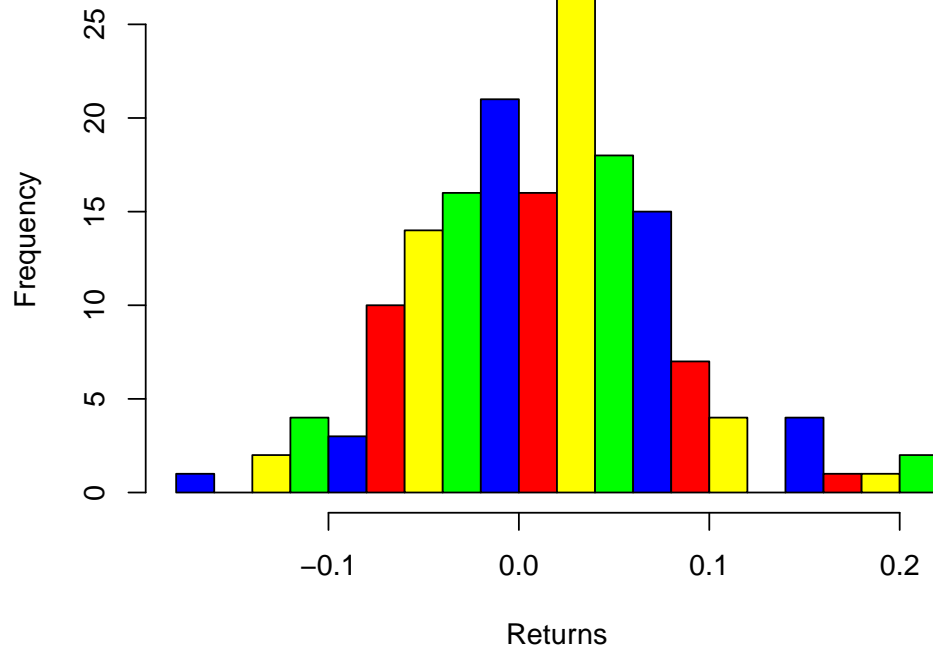
Normal Distribution Fit to Microsoft Returns

The data seem to fit a normal distribution with a mean of

```
unname(norm$estimate['mean'])
```

```
## [1] 0.01273337
```

and a standard deviation of

```
unname(norm$estimate['sd'])
```

```
## [1] 0.06472566
```

To find the probability that a monthly return will be greater than 10%, the pnorm function can be used to give us a probability of:

```
1 - pnorm(.1, mean=unname(norm$estimate['mean']), sd=unname(norm$estimate['sd']))
```

```
## [1] 0.08878835
```

4 (a)

$$\text{Find } c$$

$$f(x, y) = cx(y - x)e^{-y}, 0 \le x \le y < \infty$$

$$\int_0^\infty \int_0^y cx(y - x)e^{-y}dxdy = 1$$

$$= \int_0^\infty ce^{-y} \int_0^y yx - x^2 dxdy = 1$$

$$= \int_0^\infty ce^{-y}((\frac{1}{2}yx^2 - \frac{1}{3}x^3)\Big|_0^y)dy = 1$$

$$= \int_0^\infty ce^{-y}((\frac{1}{2}y^3 - \frac{1}{3}y^3))dy = 1$$

$$= \int_0^\infty ce^{-y}\frac{y^3}{6}dy = 1$$

$$= \frac{c}{6}(y^3(-e^y) - 3(y^2e^{-y} - 2(y(-e^{-y} - e^{-y}))))\Big|_0^\infty = 1$$

$$= 0 - (-c) = 1 = c$$

4 (b)

$$\text{Find } f_{X|Y}(x|y) \text{ and } f_{Y|X}(y|x)$$

$$f_x(x) = \int_{-\infty}^\infty F_{xy}(x, y)dy$$

$$= \int_x^\infty x(y - x)e^{-y}dy$$

$$= x(y - x)e^{-y}\Big|_x^\infty$$

$$= e^{-x}x$$

$$f_y(y) = \int_{-\infty}^\infty F_{xy}(x, y)dx$$

$$= \int_0^y x(y - x)e^{-y}dx$$

8

$$= \frac{1}{2}e^{-y}x^2y - \frac{1}{3}e^{-y}x^3 \Big|_0^y$$

$$= \frac{1}{6}e^{-y}y^3$$

$$f_{X|Y}(x|y) = \frac{f_{xy}(x,y)}{f_y(y)}$$

$$= \frac{x(y-x)e^{-y}}{\frac{1}{6}e^{-y}y^3}$$

$$f_{Y|X}(y|x) = \frac{f_{xy}(x,y)}{f_x(x)}$$

$$= \frac{x(y-x)e^{-y}}{e^{-x}x}$$

4 (c)

$$\text{Find } E(X|Y) \text{ and } E(Y|X)$$

$$E(X|Y) = \int_{-\infty}^{\infty} xf(x|y)dx$$

$$= \int_0^y x * \frac{x(y-x)e^{-y}}{\frac{1}{6}e^{-y}y^3}dx$$

$$= \frac{y}{2}$$

$$E(Y|X) = \int_{-\infty}^{\infty} yf(y|x)dy$$

$$= \int_x^{\infty} y * \frac{x(y-x)e^{-y}}{e^{-x}x}dy$$

$$= x + 2$$

5

```r
#Create a vector with 1000 max values from a sample of 10 values
#from a normal distribution with mean of 0 and standard deviation of 1
maxNorm = c()
for(i in 1:1000){
  maxNorm = append(maxNorm,max(rnorm(10, mean=0, sd=1)))
}
cat("The mean is",mean(maxNorm),"\n")
```

```
## The mean is 1.546684
```

```r
cat("The standard deviation is",sd(maxNorm),"\n")
```

```
## The standard deviation is 0.6133877
```

```r
library(ggplot2)
maxNorm = data.frame(maxNorm)
ggplot(maxNorm, aes(x=maxNorm)) +
  geom_histogram(color="#3366FF", fill="#33CCFF",binwidth = .1) +
  geom_vline(aes(xintercept=mean(maxNorm),color="Mean"),
             linetype="dashed", size=1.5,show.legend = T) +
  labs(title="Histogram of Max Values",x="Max") +
  scale_color_manual(name='',values = c(Mean = "#FF3300"))
```

Histogram of Max Values

6
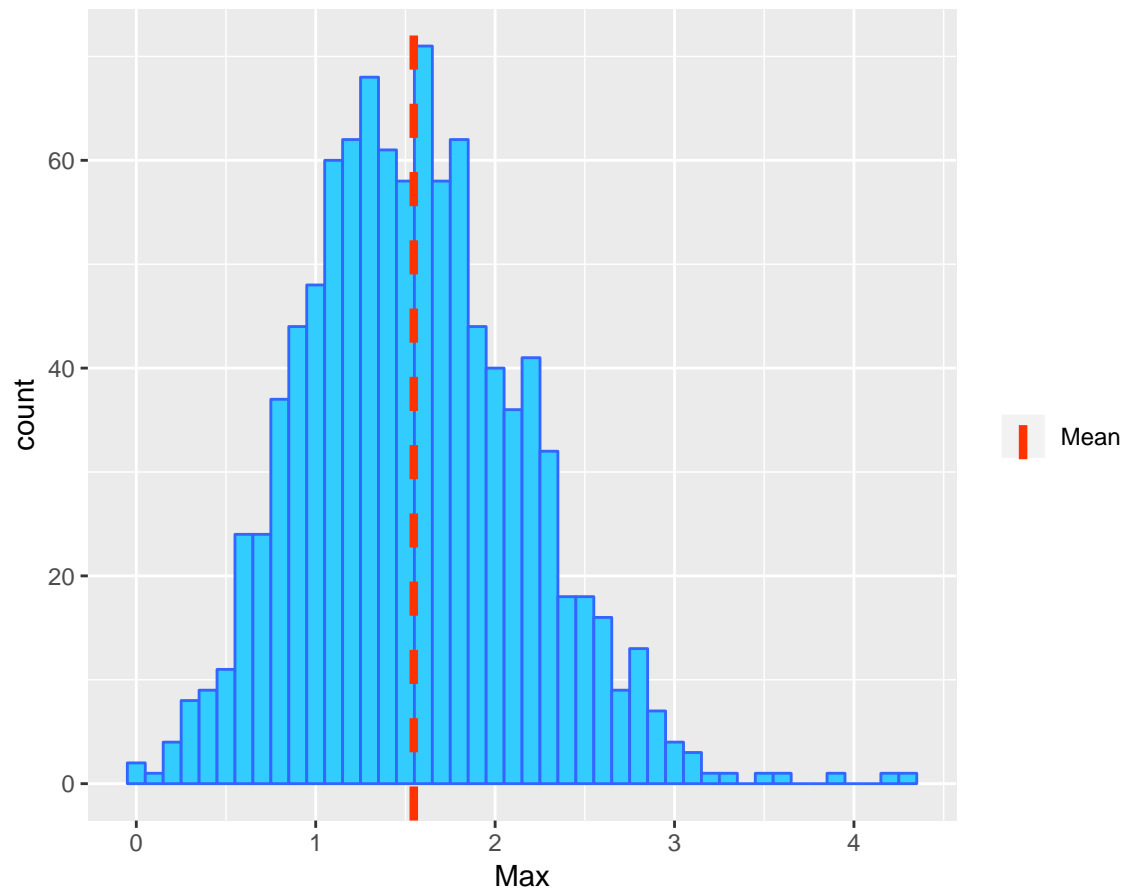
```r
#Sample 20 values from Exponential(5) and find the mean
samples = rexp(20, 5)
mean(samples)
```

```
## [1] 0.2413374
```

```r
#do the same for many samples
means = c()
for (i in 1:10000){
  samples = rexp(20, 5)
  means = append(means,mean(samples))
}
cat("The proportion of values that lie between .19 and .21 is",
    length(which(means>=.19&means<=.21))/10000,"\n")
```

```
## The proportion of values that lie between .19 and .21 is 0.184
```

```r
#do the same for n = 50
means = c()
for (i in 1:10000){
  samples = rexp(50, 5)
  means = append(means,mean(samples))
```

```
}
cat("The proportion of values that lie between .19 and .21 is",
    length(which(means>=.19&means<=.21))/10000,"\n")
```

The proportion of values that lie between .19 and .21 is 0.2761

Here we see The Weak Law of Large Numbers: We observe the average values M_n tending toward $E(X)$, for large n.

7

We know the mean of an exponential distribution is

$$\lambda^{-1}$$

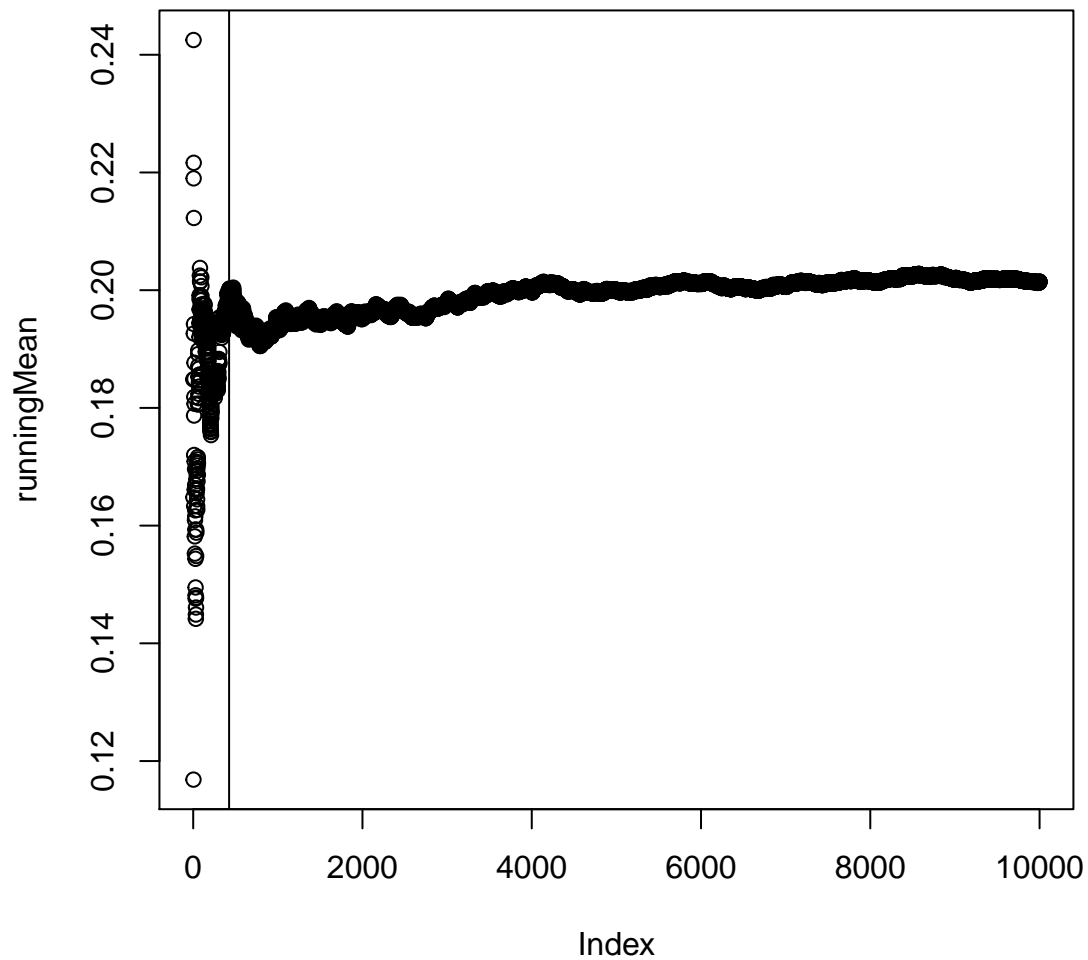so we will expect the mean to converge to 0.2. We will use

$$\epsilon = .0001$$

To test the convergence

```
#Sample 10000 values from Exponential(5) and find M1,..,Mn
set.seed(100)
samples = rexp(10000, 5)
runningMean = c()
for(i in 1:length(samples)){
  runningMean = append(runningMean,mean(samples[1:i]))
}
firstConv = min(which(abs(runningMean-.2) < .0001))

plot(runningMean)
abline(v=firstConv)
```

```r
print(firstConv)
```

```
## [1] 424
```

Here we observe the values of $M_1,\ldots,M_n$ converging to 0.2. The running mean minus the actual mean is less than epsilon after about 400 observations, but the convergence becames much clearer and stays within epsilon after roughly 3000 observations were used in calculating the mean.

8

```r
#generate 10000 means for samples of 20 values
#from a binomial distribution with size of 10 and probability of .01
samples = c()
for(i in 1:10000){
  samples = append(samples,mean(rbinom(20,10,.01)))
}
```

```
#display the means in a histogram
library(ggplot2)
samples = data.frame(samples)
ggplot(samples, aes(x=samples)) +
  geom_histogram(color="#3366FF", fill="#33CCFF",binwidth = .01) +
  labs(title="Histogram of Means")
```

Histogram of Means
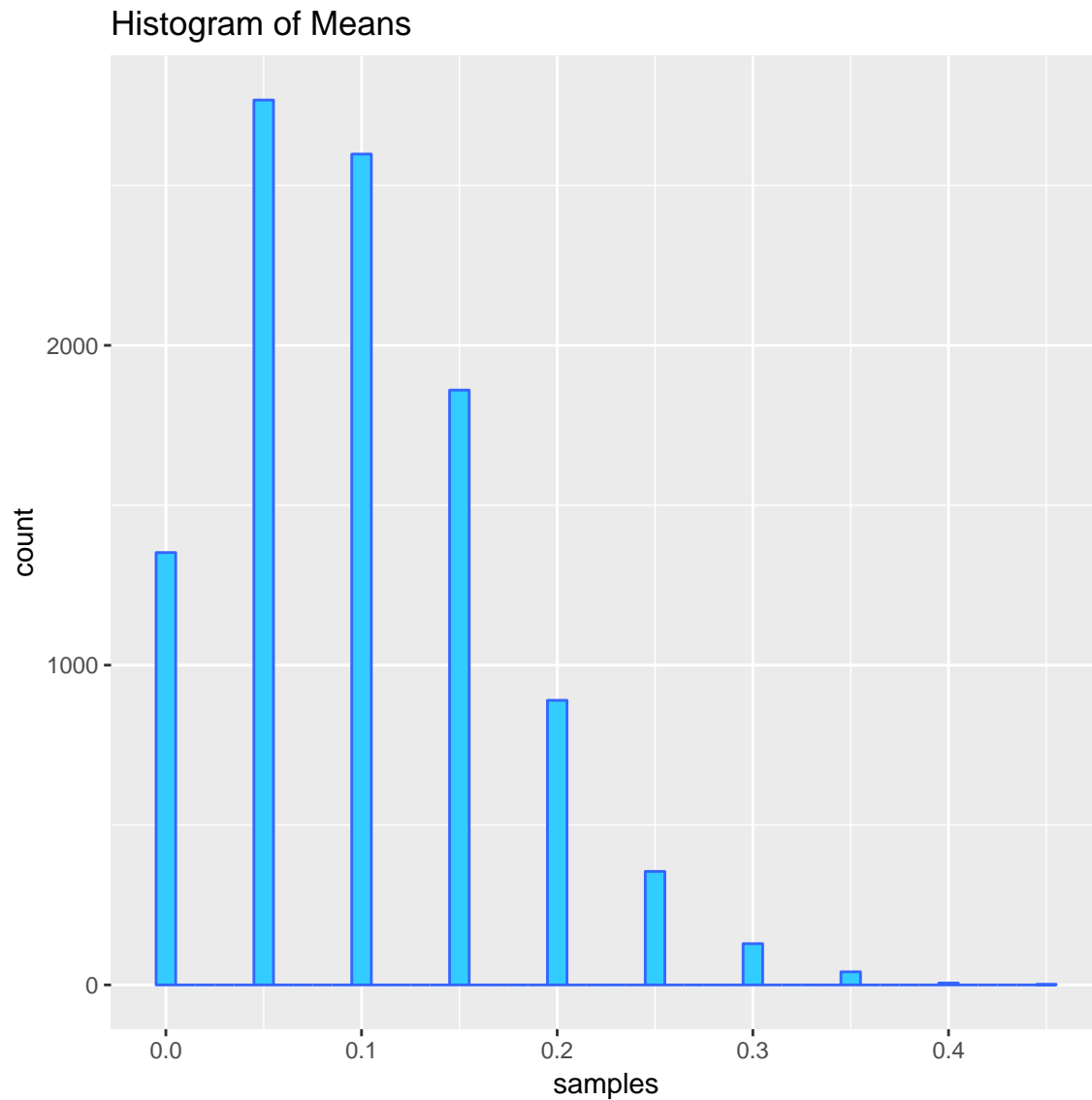

```
#graphical description of the distribution
sampVec = samples$samples
descdist(sampVec, boot = 1000)
```

Cullen and Frey graph
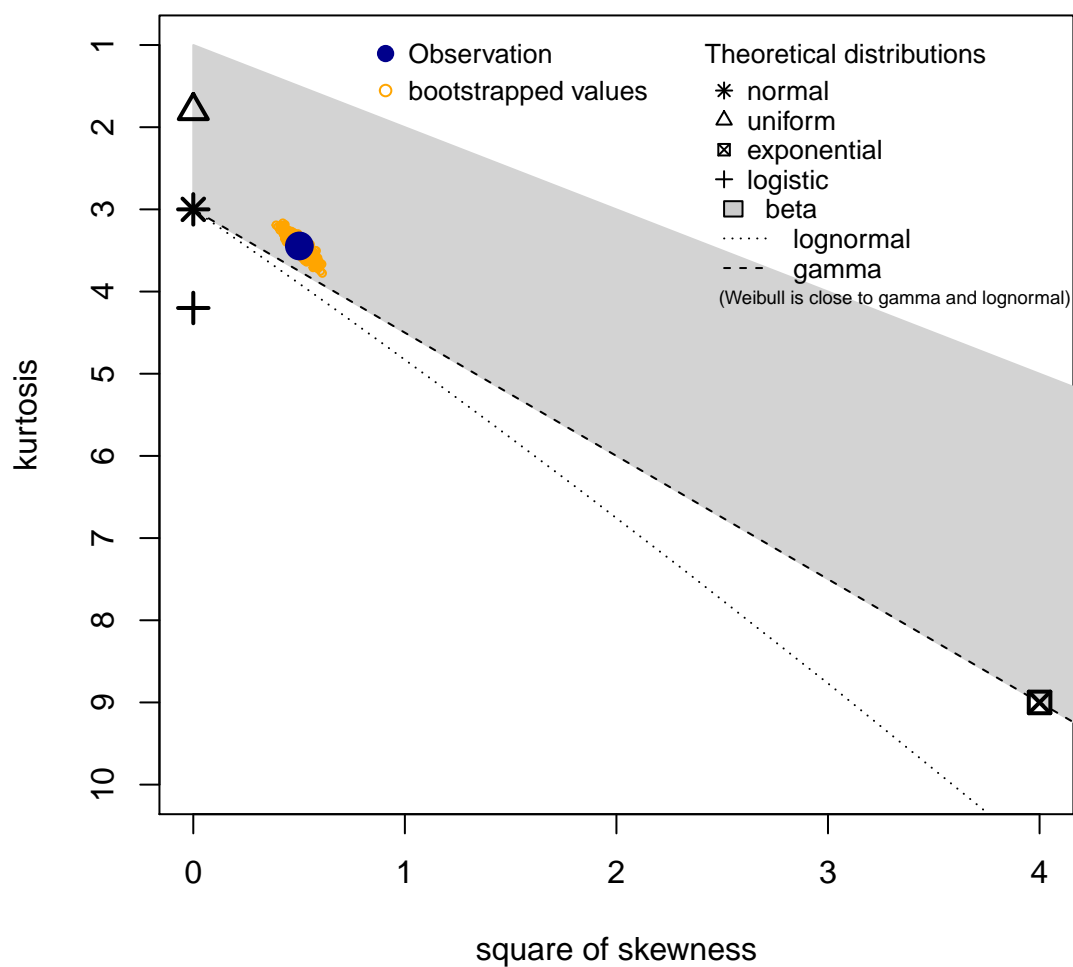

```
## summary statistics
## ------
## min:  0    max:  0.45
## median:  0.1
## mean:  0.100025
## estimated sd:  0.07105459
## estimated skewness:  0.7080674
## estimated kurtosis:  3.445038
```

This looks like a gamma distribution or a beta distribution rather than a normal distribution.