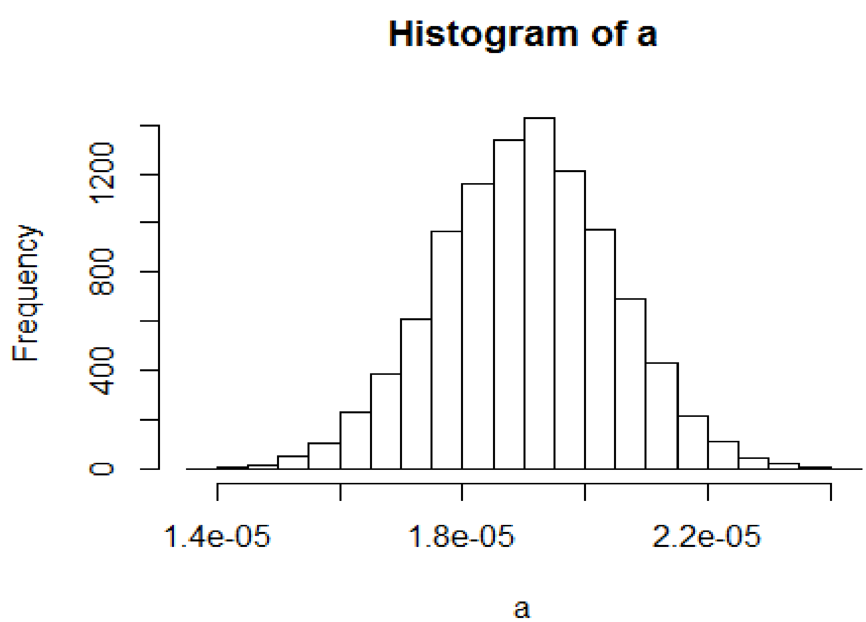
LAB #4

Dheeraj Allamaneni

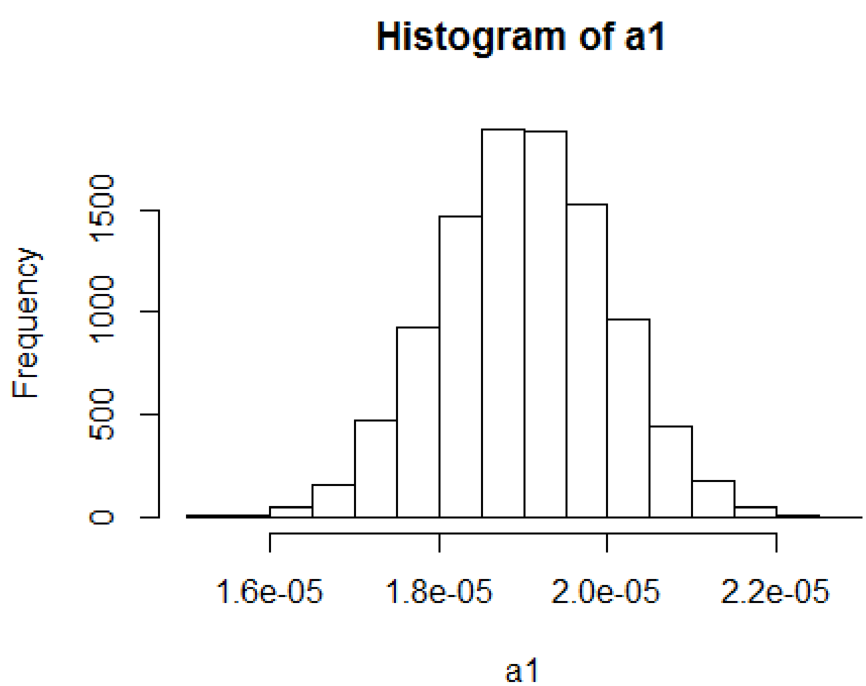
20 September 2018

Question 1

```
#this function alpha is commonly used for all the below computing
alpha<-function(l1,l2,t1,t2){
  (l2-l1)/(l1*(t2-t1))
}

L1<-rnorm(10000,mean=1,sd=.00005)#initial length
L2<-rnorm(10000,mean=1.00095,sd=.00005)#Final length
T1<-rnorm(10000,mean=50,sd=.1)#initial Temp
T2<-rnorm(10000,mean=100,sd=.1)#Final Temp
a<-rep(0,10000)#This repeats value 0 10000 times. this is a vector of values
0 ten thousand times

for(i in 1:10000) {a[i]<-alpha(L1[i],L2[i],T1[i],T2[i])}
summary(a)

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 1.391e-05 1.806e-05 1.905e-05 1.903e-05 2.000e-05 2.426e-05

sd(a)

## [1] 1.426974e-06

hist(a)
```

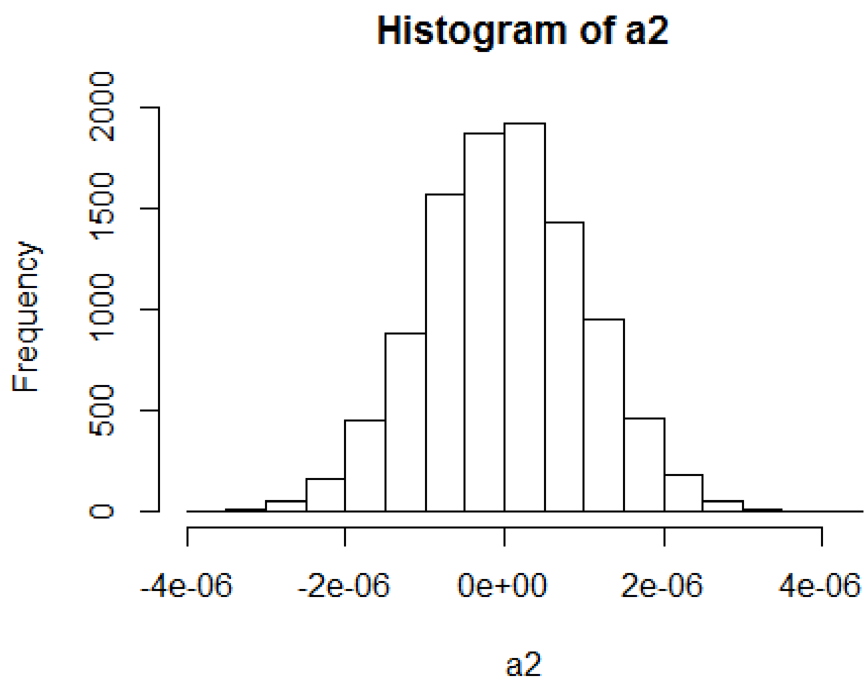


```r
#Now L1 constant
L1<-rep(1,10000)

L2<-rnorm(10000,mean=1.00095,sd=.00005)#Final length
T1<-rnorm(10000,mean=50,sd=.1)#initial Temp
T2<-rnorm(10000,mean=100,sd=.1)#Final Temp
a1<-rep(0,10000)#This repeats value 0 10000 times. this is a vector of values
0 ten thousand times

for(i in 1:10000) {a1[i]<-alpha(L1[i],L2[i],T1[i],T2[i])}
summary(a1)

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 1.507e-05 1.833e-05 1.901e-05 1.901e-05 1.969e-05 2.268e-05

sd(a1)

## [1] 1.001918e-06

hist(a1)
```



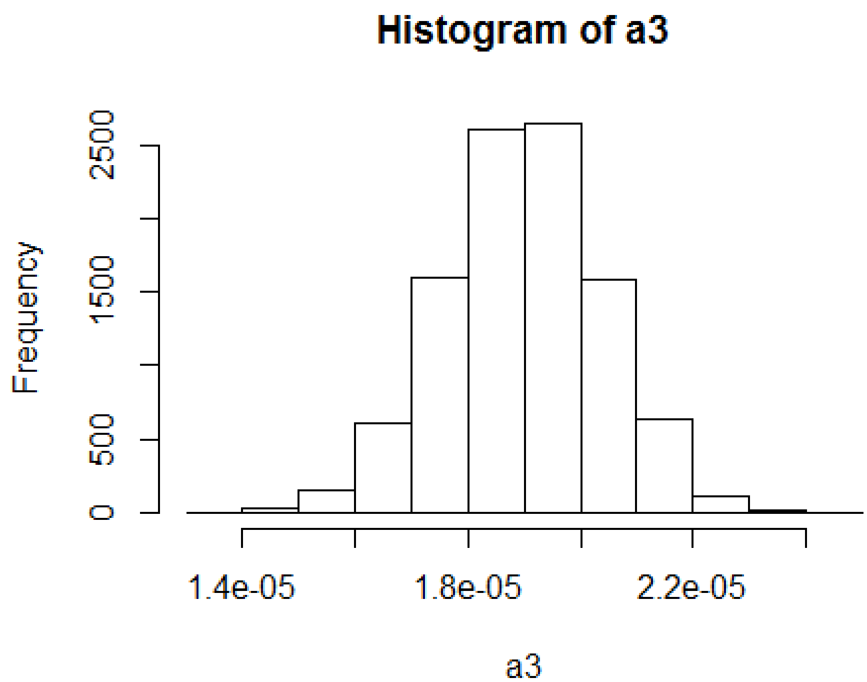

```r
#Now L2 Constant

L1<-rnorm(10000,mean=1,sd=.00005)#initial length
L2<-rep(1,10000)
T1<-rnorm(10000,mean=50,sd=.1)#initial Temp
T2<-rnorm(10000,mean=100,sd=.1)#Final Temp
a2<-rep(0,10000)#This repeats value 0 10000 times. this is a vector of values
0 ten thousand times
```

```r
for(i in 1:10000) {a2[i]<-alpha(L1[i],L2[i],T1[i],T2[i])}
summary(a2)

##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -3.907e-06 -6.869e-07  6.220e-10  1.632e-09  6.782e-07  4.411e-06

sd(a2)

## [1] 1.008271e-06

hist(a2)
```




```r
#Now T1 Constant

L1<-rnorm(10000,mean=1,sd=.00005)#initial length
L2<-rnorm(10000,mean=1.00095,sd=.00005)#Final length
T1<-rep(50,10000)
T2<-rnorm(10000,mean=100,sd=.1)#Final Temp
a3<-rep(0,10000)#This repeats value 0 10000 times. this is a vector of values
0 ten thousand times

for(i in 1:10000) {a3[i]<-alpha(L1[i],L2[i],T1[i],T2[i])}
summary(a3)

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 1.385e-05 1.805e-05 1.900e-05 1.899e-05 1.994e-05 2.437e-05

sd(a3)

## [1] 1.400829e-06
```

```r
hist(a3)
```

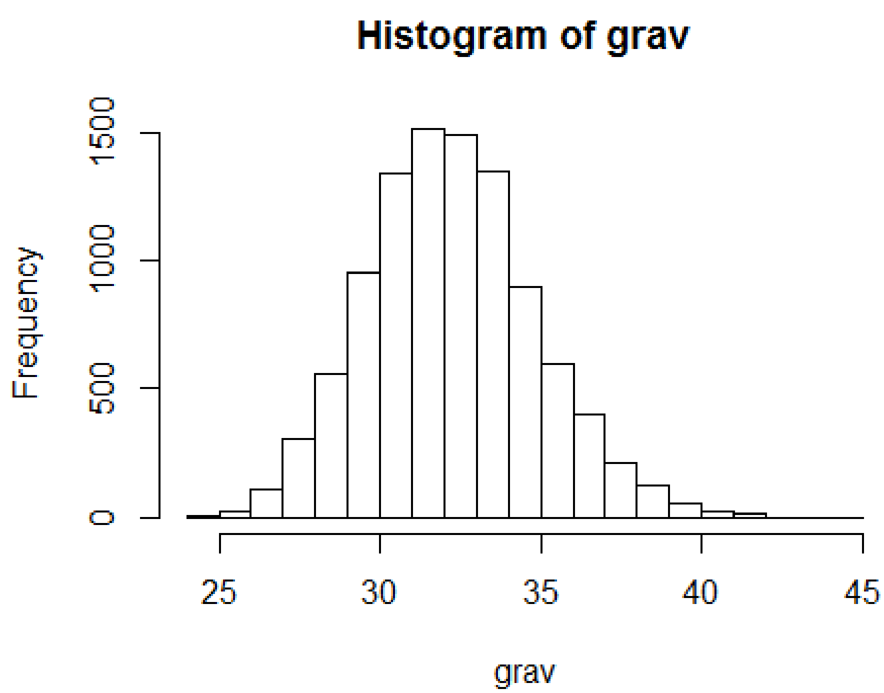



```r
#Now T2 Constant


L1<-rnorm(10000,mean=1,sd=.00005)#initial length
L2<-rnorm(10000,mean=1.00095,sd=.00005)#Final length
T1<-rnorm(10000,mean=50,sd=.1)#initial Temp
T2<-rep(100,10000)
a4<-rep(0,10000)#This repeats value 0 10000 times. this is a vector of values
0 ten thousand times

for(i in 1:10000) {a4[i]<-alpha(L1[i],L2[i],T1[i],T2[i])}
summary(a4)

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 1.351e-05 1.804e-05 1.897e-05 1.899e-05 1.996e-05 2.470e-05

sd(a4)

## [1] 1.413202e-06

#COefficient of linear expansion of brass is with mean
mean(a)

## [1] 1.903227e-05

# and standard deviation of
sd(a)

## [1] 1.426974e-06
```

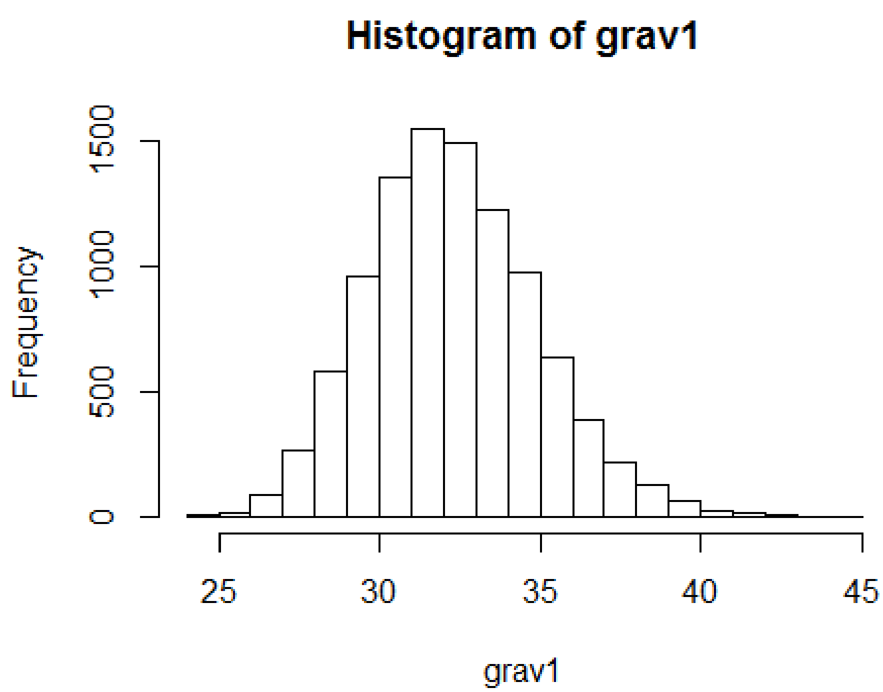
```
#Temperature(primarily T1) is major cause of uncertainity when compared to to
Length
#The coefficient with the changing parameters produces a greater standard dev
iation when compared to the distributed parameters. The nonlinearity of the c
oefficient is a factor for this.
```

Question 2

```
gravityfun<-function(l1,t1){
  ((2*pi)^2 * l1)/(t1^2)
}

L1<-rnorm(10000,mean=5,sd=0.0208)#Length

T1<-rnorm(10000,mean=2.48,sd=.1)#Time Period
grav<-rep(0,10000)#This repeats value 0 10000 times. this is a vector of valu
es 0 ten thousand times

for(i in 1:10000) {grav[i]<-gravityfun(L1[i],T1[i])}
summary(grav)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   24.32   30.44   32.12   32.25   33.87   44.11

sd(grav)

## [1] 2.62342

hist(grav)
```

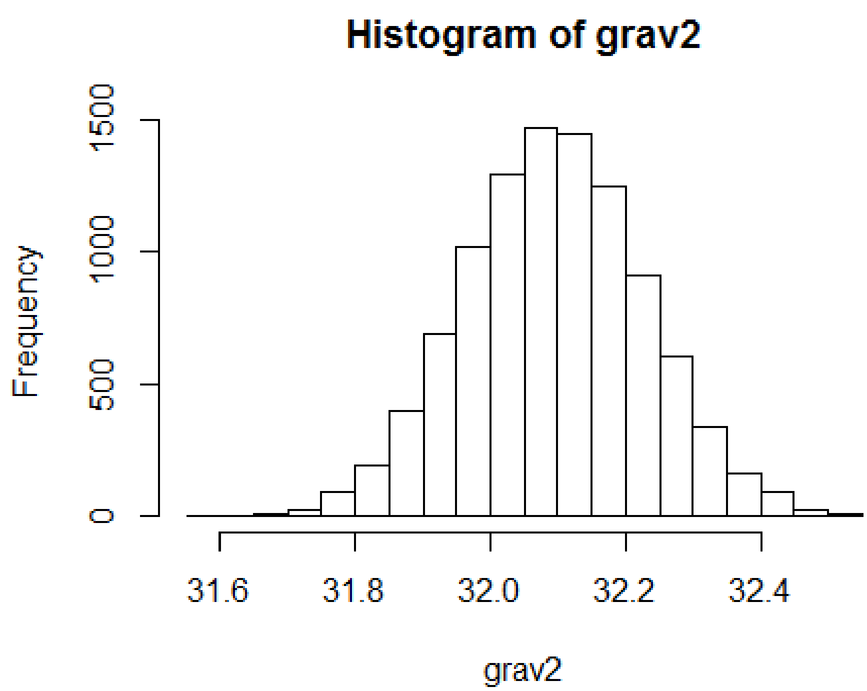


```r
#NOW L constant

L1<-rep(5,10000)

T1<-rnorm(10000,mean=2.48,sd=.1)#Time Period
grav1<-rep(0,10000)#This repeats value 0 10000 times. this is a vector of val
ues 0 ten thousand times

for(i in 1:10000) {grav1[i]<-gravityfun(L1[i],T1[i])}
summary(grav1)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   24.00   30.45   32.12   32.29   33.96   44.31

sd(grav1)

## [1] 2.623708

hist(grav1)
```




```r
#NOW Time period const

gravityfun<-function(l1,t1){
  ((2*pi)^2 * l1)/(t1^2)
}

L1<-rnorm(10000,mean=5,sd=0.0208)#Length
```

```r
T1<-rep(2.48,10000)
grav2<-rep(0,10000)#This repeats value 0 10000 times. this is a vector of val
ues 0 ten thousand times

for(i in 1:10000) {grav2[i]<-gravityfun(L1[i],T1[i])}
summary(grav2)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   31.59   32.00   32.09   32.09   32.18   32.55

sd(grav2)

## [1] 0.1331717

hist(grav2)
```

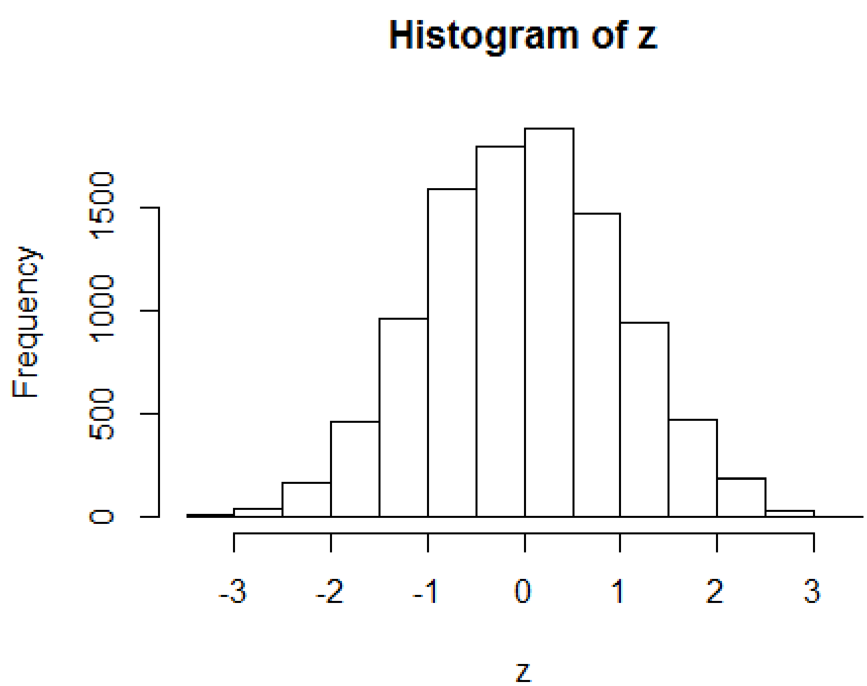



```r
#The g value comes out to be
mean(grav)

## [1] 32.25277

#and the standard deviation is
sd(grav)

## [1] 2.62342
```

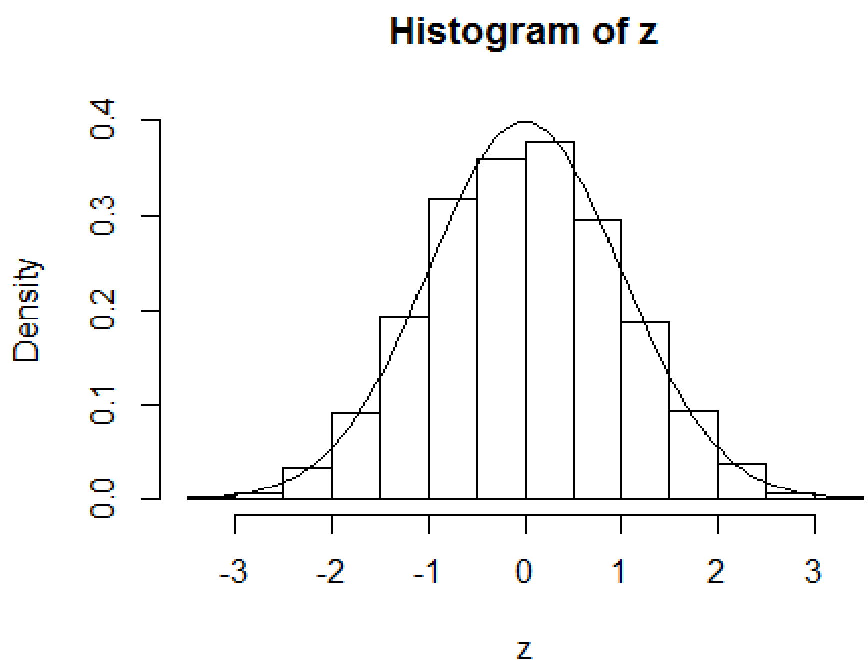
```
#IN this problem both L and Time period show the similar contribution to the
value g.
#The actual values of g varies from 32.09ft/sec2 and 32.26ft/sec2
```

Question 3

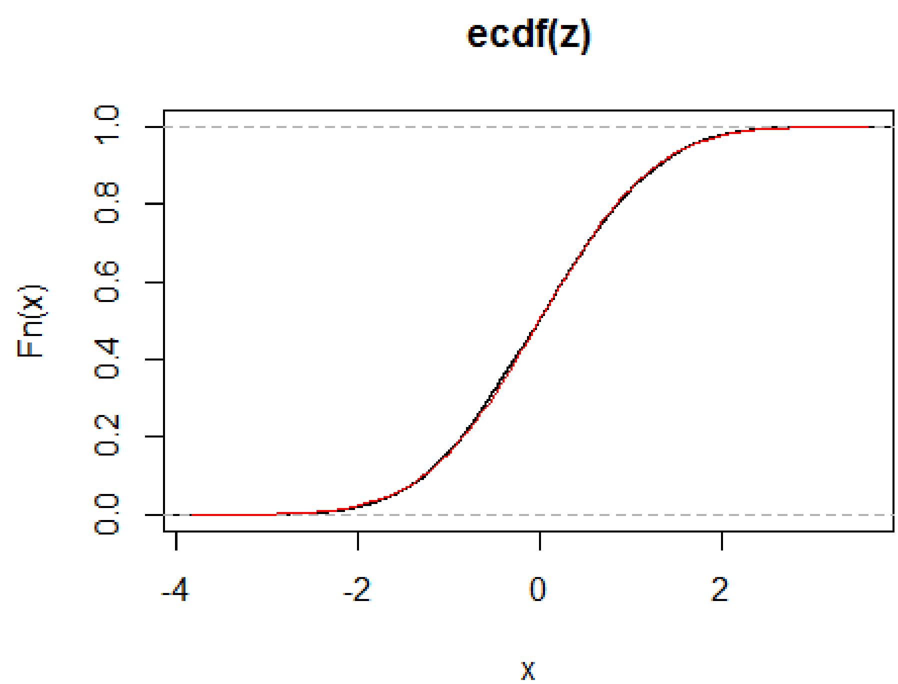
```
# N = 5
M<-matrix(runif(50000,min=0,max=1),nrow=10000,byrow=T)

av<-1:10000

for (i in 1:10000){
  av[i]<-mean(M[i,])
}

z<-1:10000
for (i in 1:10000){
  z[i]<-(av[i]-.5)*sqrt(60)
}

r<-range(0,hist(z)$density,dnorm(0,sd=1))
```




```
hist(z,freq=FALSE,ylim=r)
curve(dnorm(x,mean=0,sd=1),add=TRUE)
```

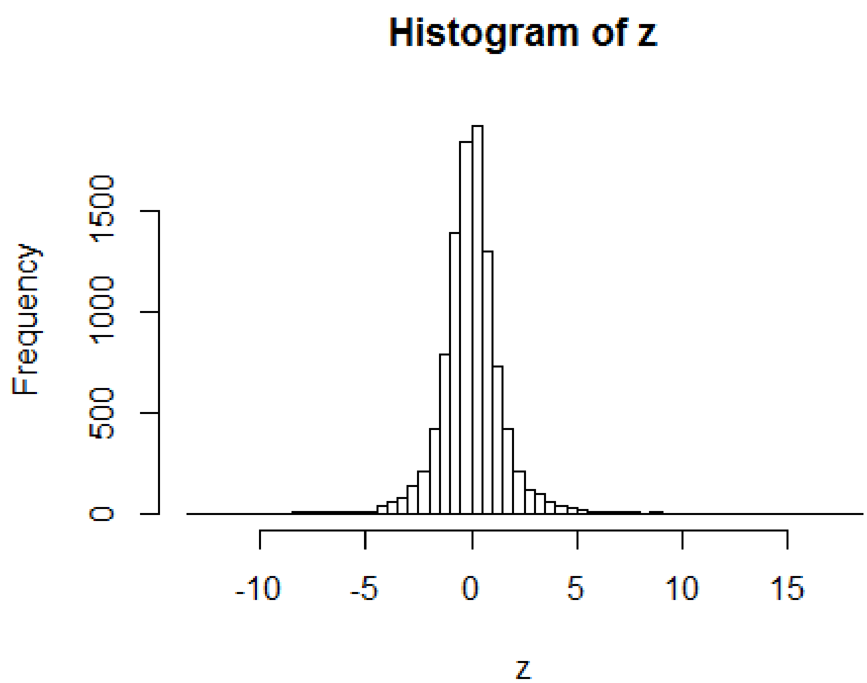



```
plot(ecdf(z))
curve(pnorm(x),add=TRUE,col="red")
```

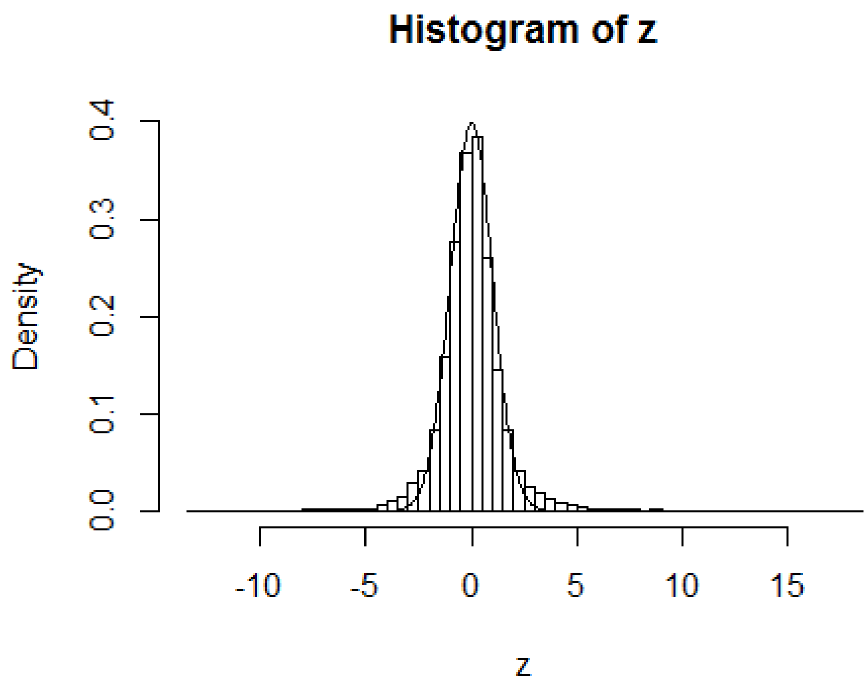



```
#Now use the sample standard deviation rather than the model sigma=1/sqrt(12)
#to make "z" values and see that these don't look as much like standard norma
l
```

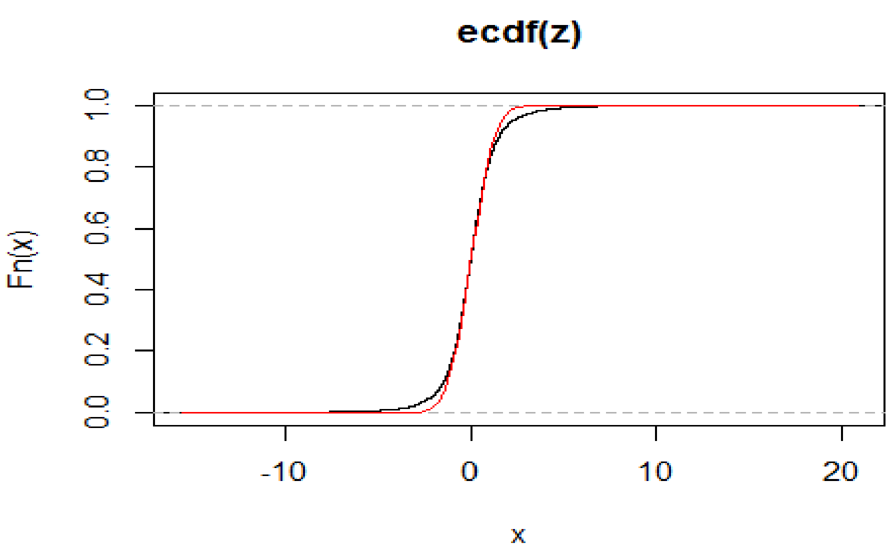
```
#observations as do the properly standardized values of xbar

s<-1:10000
for (i in 1:10000){
  s[i]<-sd(M[i,])
}

z<-1:10000
for (i in 1:10000){
  z[i]<-(av[i]-.5)*sqrt(5)/s[i]
}

r<-range(0,hist(z,breaks=100)$density,dnorm(0,sd=1))
```



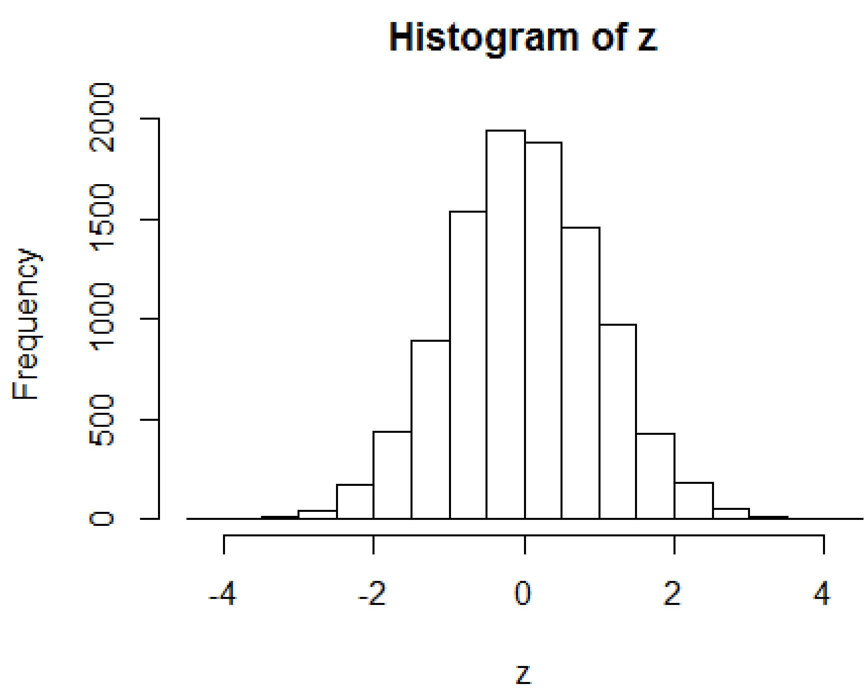

```
#the breaks=100 changes the default number of histogram bins so we can
#see some detail in the center of the distribution

hist(z,breaks=100,freq=FALSE,ylim=r)
curve(dnorm(x,mean=0,sd=1),n=500,add=TRUE)
```



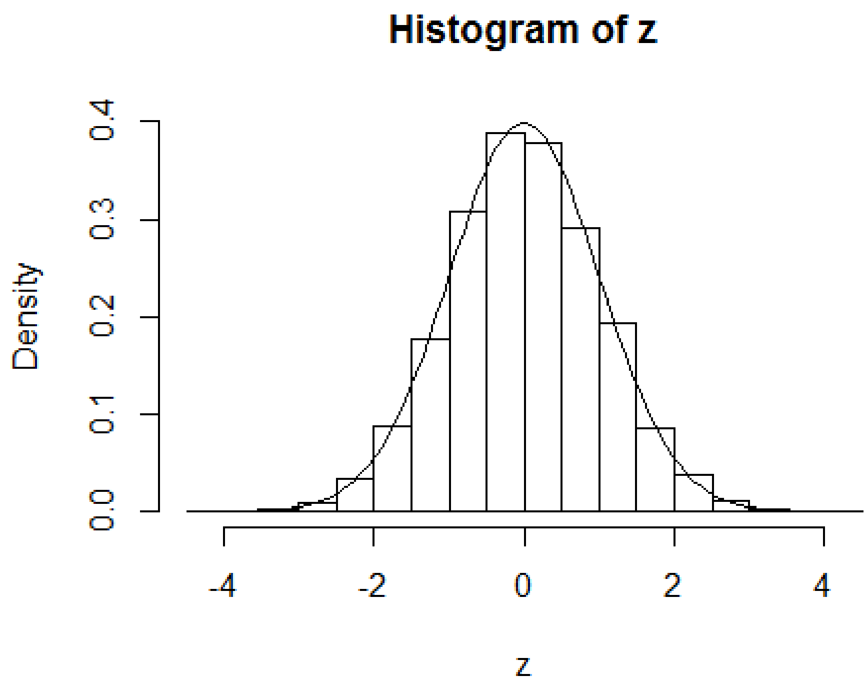

```
#the n=500 provides enough plotted points (connected with line segments)
#to produce a nice plot near the mean of the distribution

summary(z)

##        Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -13.150194  -0.707518  -0.003062  -0.001458   0.685202  18.381272

plot(ecdf(z))
curve(pnorm(x),add=TRUE, col="red")
```



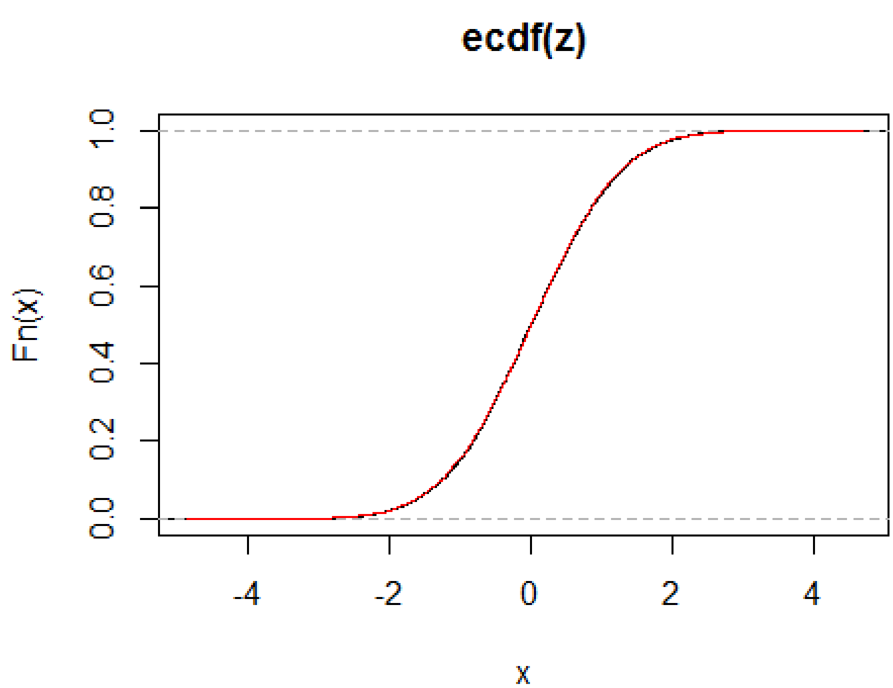
```
#distribution of z^ are more spreadout and z are more closely like std normal
s

################################################

#  N=100

M<-matrix(runif(1000000,min=0,max=1),nrow=10000,byrow=T)

av<-1:10000

for (i in 1:10000){
  av[i]<-mean(M[i,])
}

z<-1:10000
for (i in 1:10000){
  z[i]<-(av[i]-.5)*sqrt(1200)
}

r<-range(0,hist(z)$density,dnorm(0,sd=1))
```



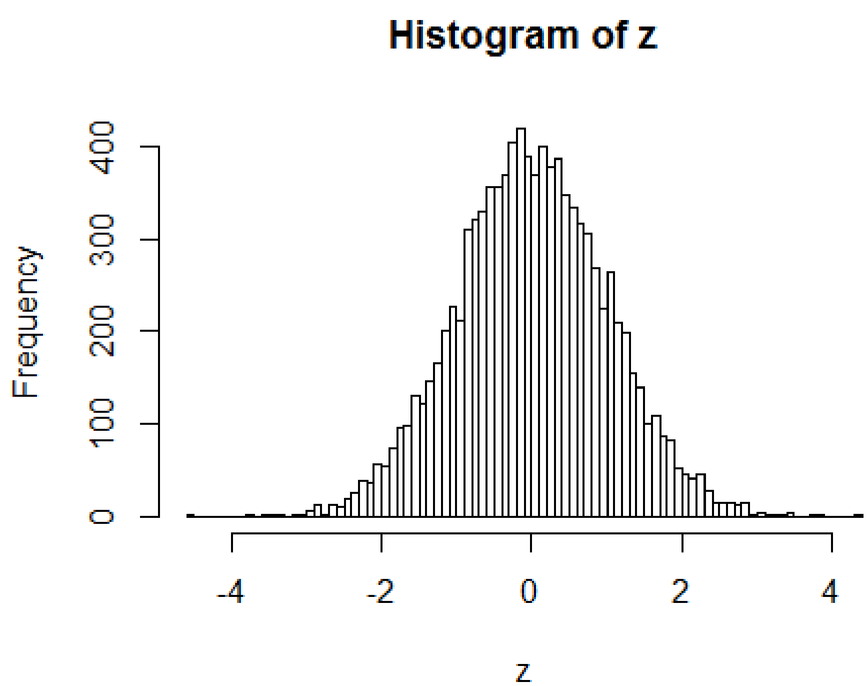

```
hist(z,freq=FALSE,ylim=r)
curve(dnorm(x,mean=0,sd=1),add=TRUE)
```



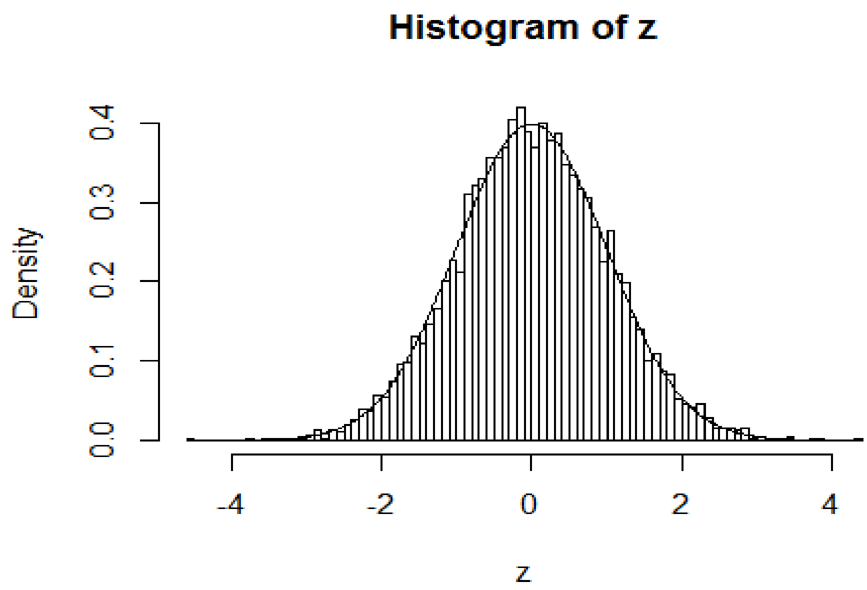

```r
plot(ecdf(z))
curve(pnorm(x),add=TRUE,col="red")
```



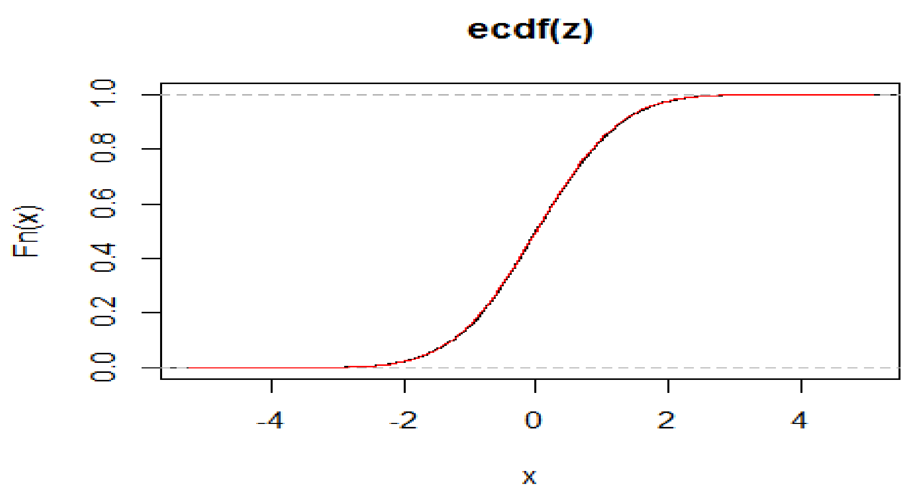
```
#Now use the sample standard deviation rather than the model sigma=1/sqrt(12)
#to make "z" values and see that these don't look as much like standard norma
l
#observations as do the properly standardized values of xbar

s<-1:10000
for (i in 1:10000){
  s[i]<-sd(M[i,])
}

z<-1:10000
for (i in 1:10000){
  z[i]<-(av[i]-.5)*sqrt(100)/s[i]
}

r<-range(0,hist(z,breaks=100)$density,dnorm(0,sd=1))
```




```
#the breaks=100 changes the default number of histogram bins so we can
#see some detail in the center of the distribution

hist(z,breaks=100,freq=FALSE,ylim=r)
curve(dnorm(x,mean=0,sd=1),n=500,add=TRUE)
```




```
#the n=500 provides enough plotted points (connected with line segments)
#to produce a nice plot near the mean of the distribution

summary(z)

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -4.533484 -0.662571 -0.005298  0.007563  0.682023  4.361312

plot(ecdf(z))
curve(pnorm(x),add=TRUE, col="red")
```




```
#Sample Mean dstn for the given data stanard deviation is Standard Normal dis
tribution. Where as with the
#Known sample standard deviation is not. Mainly beacuse N = 5 is a very small
sample size
#Where as in N=100  the sample mean distribution and standard deviation are s
```

```
tandard normal because 100 is quite a large sample.

################################################
```

Question 4

```
#First case N = 5

M<-matrix(runif(50000,min=0,max=1),nrow=10000,byrow=T)
Low<-rep(0,10000)
Up<-rep(0,10000)

chk<-rep(0,10000)
for (i in 1:10000){
  av[i]<-mean(M[i,])
}

for(i in 1:10000) {Low[i]<-av[i]-1.96*sqrt(1/60)}
for(i in 1:10000) {Up[i]<-av[i]+1.96*sqrt(1/60)}
for(i in 1:10000) {if((Low[i]<.5)&(.5<Up[i])) chk[i]<-1}

cbind(Low[1:10],Up[1:10],chk[1:10])

##             [,1]      [,2] [,3]
##  [1,] 0.2907083 0.7967781    1
##  [2,] 0.3617736 0.8678434    1
##  [3,] 0.4591847 0.9652545    1
##  [4,] 0.2886904 0.7947602    1
##  [5,] 0.1502292 0.6562991    1
##  [6,] 0.1968299 0.7028997    1
##  [7,] 0.2534082 0.7594780    1
##  [8,] 0.3661795 0.8722494    1
##  [9,] 0.2064336 0.7125034    1
## [10,] 0.2582648 0.7643347    1

mean(chk)

## [1] 0.9482

#Now use the sample standard deviation to make intervals for mu

Low<-rep(0,10000)
Up<-rep(0,10000)
chk<-rep(0,10000)
s<-rep(0,10000)
for (i in 1:10000){
  s[i]<-sd(M[i,])
}

for(i in 1:10000) {Low[i]<-av[i]-1.96*s[i]*sqrt(1/5)}
for(i in 1:10000) {Up[i]<-av[i]+1.96*s[i]*sqrt(1/5)}
```

```r
for(i in 1:10000) {if((Low[i]<.5)&(.5<Up[i])) chk[i]<-1}

cbind(Low[1:10],Up[1:10],chk[1:10])

##            [,1]      [,2] [,3]
##  [1,] 0.2458955 0.8415909    1
##  [2,] 0.4475136 0.7821033    1
##  [3,] 0.5291555 0.8952836    0
##  [4,] 0.1807373 0.9027133    1
##  [5,] 0.1160313 0.6904970    1
##  [6,] 0.2855290 0.6142006    1
##  [7,] 0.1665007 0.8463855    1
##  [8,] 0.3115139 0.9269150    1
##  [9,] 0.2581370 0.6608000    1
## [10,] 0.2646811 0.7579184    1

mean(chk)

## [1] 0.8643

#Second N=100


M<-matrix(runif(1000000,min=0,max=1),nrow=10000,byrow=T)
Low<-rep(0,10000)
Up<-rep(0,10000)

chk<-rep(0,10000)
for (i in 1:10000){
  av[i]<-mean(M[i,])
}

for(i in 1:10000) {Low[i]<-av[i]-1.96*sqrt(1/1200)}
for(i in 1:10000) {Up[i]<-av[i]+1.96*sqrt(1/1200)}
for(i in 1:10000) {if((Low[i]<.5)&(.5<Up[i])) chk[i]<-1}

cbind(Low[1:10],Up[1:10],chk[1:10])

##            [,1]      [,2] [,3]
##  [1,] 0.4300872 0.5432479    1
##  [2,] 0.4064518 0.5196125    1
##  [3,] 0.3951987 0.5083594    1
##  [4,] 0.4131003 0.5262610    1
##  [5,] 0.4247333 0.5378940    1
##  [6,] 0.4197400 0.5329007    1
##  [7,] 0.4494634 0.5626240    1
##  [8,] 0.4209814 0.5341420    1
##  [9,] 0.4166435 0.5298042    1
## [10,] 0.4382926 0.5514533    1

mean(chk)
```

```
## [1] 0.9492

#Now use the sample standard deviation to make intervals for mu

Low<-rep(0,10000)
Up<-rep(0,10000)
chk<-rep(0,10000)
s<-rep(0,10000)
for (i in 1:10000){
  s[i]<-sd(M[i,])
}

for(i in 1:10000) {Low[i]<-av[i]-1.96*s[i]*sqrt(1/100)}
for(i in 1:10000) {Up[i]<-av[i]+1.96*s[i]*sqrt(1/100)}
for(i in 1:10000) {if((Low[i]<.5)&(.5<Up[i])) chk[i]<-1}

cbind(Low[1:10],Up[1:10],chk[1:10])

##            [,1]      [,2] [,3]
##  [1,] 0.4301561 0.5431790    1
##  [2,] 0.4055704 0.5204939    1
##  [3,] 0.3982076 0.5053505    1
##  [4,] 0.4135288 0.5258324    1
##  [5,] 0.4311932 0.5314341    1
##  [6,] 0.4203759 0.5322648    1
##  [7,] 0.4479774 0.5641101    1
##  [8,] 0.4188625 0.5362609    1
##  [9,] 0.4180100 0.5284377    1
## [10,] 0.4354432 0.5543026    1

mean(chk)

## [1] 0.9496

#Target confidence level is 95% and the calculated confidence level is 95.42%
; But for the sample standard deviation we get confidence value or level as 8
6.47 for n= 5 as the sample size is very small.
#For N = 100Target confidence level is 95% and confidence level calculated is
94.92 (very close).
#known sample SD confidence level as 94.96 for n=100 ; as the sample is large
enough.
```