Assignment

Matthew Mariano

February 21, 2015

Introduction

The purpose of this project is to explore the Central Limit Theorem using the exponential distribution. The CLT allows us to investigate the mean of a distribution given samples from that distribution. The CLT gives us the expected mean and variance for the distibution of sample mean in the limit as the number of samples goes to infinity. In other words if we take a sample of size n from a distribution then it has a mean called the sample mean. If we collect these sample means from many random simulations then we have a new distribution which is just a distribution of sample means , where the sample size is n.

The CLT states that the mean of distribution just described has the mean of the original population with a variance of the original population divided by n.

The Equation for the Normal or Guassian Distribution

$$\frac{e^{-}((X-\mu)/(2\sigma^2))}{\left(\sigma\sqrt{2\pi}\right)}$$

- X is a random variable
- sigma is the standard deviation
- mu is the mean

$$f(x,\lambda) = \begin{cases} exp(-\lambda x) & x \geq 0 \\ 0 & x < 0 \end{cases}$$

- lambda is called the rate paramter.
- the mean is 1/lambda
- the standard deviation is 1/lambda

First define some variables and set the seed for repeatability.

- nosims = number of simulations
- nsamples = the sample size
- lambda = the rate parameter associated with the exponential distribution
- xmu = the expected mean for the distribution of sample means
- xsd = the expected standard deviation for the distribution of sample means
- psd is the original population(from the exponenntial distribution) standard deviation

```
set.seed(123)
nsamples=40
nosims=1000
lambda=.2;
xmu=1/lambda
xsd=(1/lambda)/sqrt(nsamples)
psd=1/lambda
```

The function getdist returns a matrix containing the simulation id , distribution of means and the distribution of standard deviations for our simulations.

```
getdist=function(){
  dist=rexp(nosims*nsamples,rate=lambda)
  v1=c(1:nosims)
  v2=apply(matrix(data=dist,nrow=nosims),1,mean)
  v3=apply(matrix(data=dist,nrow=nosims),1,sd)
  m=cbind(v1,v2,v3)
  dimnames(m)=list(c(),c("id","mean","sd"))
  m
}
```

Call getdist to get the distributions. Use the mean of the standard deviation from all of the simulations and compare with the expected standard deviation for the original population, 1/lambda. Then use the mean of the means and compare it to the expected value of 1/lambda

- means is the distribution of means
- nmeans is the normallized distribution of means
- sds = the distribution of standard deviations
- msd = the mean of the standard deviations
- mm = the mean of the means

```
m=getdist()
means=m[,c("mean")]
nmeans=(m[,c("mean")]-xmu)/xsd
sds=m[,c("sd")]
msd=mean(sds)
pdiff=abs(msd -  psd)/(msd+psd)*200
message("percentage difference for sigma=",pdiff)
```

```
## percentage difference for sigma=2.49715552347027
```

```
mm=mean(means)
pdiff2=abs(mm -  xmu)/(mm+xmu)*200
message("percentage difference for mu=",pdiff2)
```

```
## percentage difference for mu=0.237942152907709
```

Next define functions for collecting the mean of means over many simulations and varying sample sizes.

```
varyingDist1=function(n,start){
  nseq=seq(start,by=1,length.out=n)
  mmatrix=matrix(nrow = nosims, ncol=n)
  nsamples1=start
  nsamples2=nsamples1+n-1
  v=vector(length=n)
  for(nsamples in nsamples1:nsamples2){
     dist=rexp(nosims*nsamples,rate=lambda)
     v2=apply(matrix(data=dist,nrow=nosims),1,mean)
     mm=mean(v2)
     pdiff=abs(mm-xmu)/(mm+xmu)*200
     v[nsamples-nsamples1+1]=pdiff
  }
  v
}
```

Next define functions for drawing the plots which will appear in the appendix. Not the use of freq=F to gives a histogram of area 1 and allows us to easily overlay a normal distribution.

```
plot1=function(){
   hist(means,breaks=200,xlim=xmu+c(-1,1)*3*xsd,freq=F,main="The Distribution of Sample Means")
   xfit=as.vector(seq(from=min(v),to=max(means),length.out=nosims))
   xfit2=(xfit-xmu)/xsd
   yfit=1/(xsd*sqrt(2*pi))*exp(-xfit2^2/2)
   lines(x=xfit,y=yfit,type ="l",lwd = 2 ,col="red")
   x=c(mm,mm)
   y=c(0,dnorm(mm,mean=xmu,sd=xsd))
   lines(x=x,y=y,lwd=5,col="blue")
   x=xmu+c(-2,2)*xsd
   y=dnorm(x,mean=xmu,sd=xsd);
   lines(x=x,y=y,lwd=5,col="blue")
}
plot2=function(){

  hist(nmeans,xlim=c(-1,1)*3,breaks=100,main="Normallized Distribution of Means",freq=F)
  xfit=seq(-3,3,length=nosims*nsamples)
  yfit=1/sqrt(2*pi)*exp(-xfit^2/2)
  lines(x=xfit,y=yfit,type ="l",lwd = 1 ,col = "red")
  x=c(0,0)
  y=c(0,dnorm(0))
  lines(x=x,y=y,lwd=5,col="blue")
  x=c(-2,2)
  y=dnorm(x);
  lines(x=x,y=y,lwd=5,col="blue")
}

#
# show that the sample stadndard deviation follows a sigma/sqrt(n)
#
# parm n - the number of samples sizes
# start is the starting sample size
plot3=function(n,start){
  x=seq(start,by=1,length.out=n)
  mmatrix=matrix(nrow = nosims, ncol=n)
  nsamples1=start
  nsamples2=nsamples1+n-1
  v=vector(length=n)
  for(nsamples in nsamples1:nsamples2){
     dist=rexp(nosims*nsamples,rate=lambda)
     v2=apply(matrix(data=dist,nrow=nosims),1,mean)
     sd1=sd(v2)
     sd2=psd/sqrt(nsamples)
     v[nsamples-nsamples1+1]=sd1
  }
  number_samples=x
  sigma_n=v
  plot(number_samples,sigma_n)
  y=psd/sqrt(x)
  lines(x,y,lwd=5,col="blue")
}
```

Appendix

Figure 1 The Distribution of sample means.

- normal distribution with mu=1/lambda and sigma=1/lambda is overlayed.
- The vertical blue line is drawn at the mean which is almost exactly the expected mean of 5.
- The horizontal blue line is drawn at plus and minus 2 standard deviations and covers the interval 3.42 to 6.58.

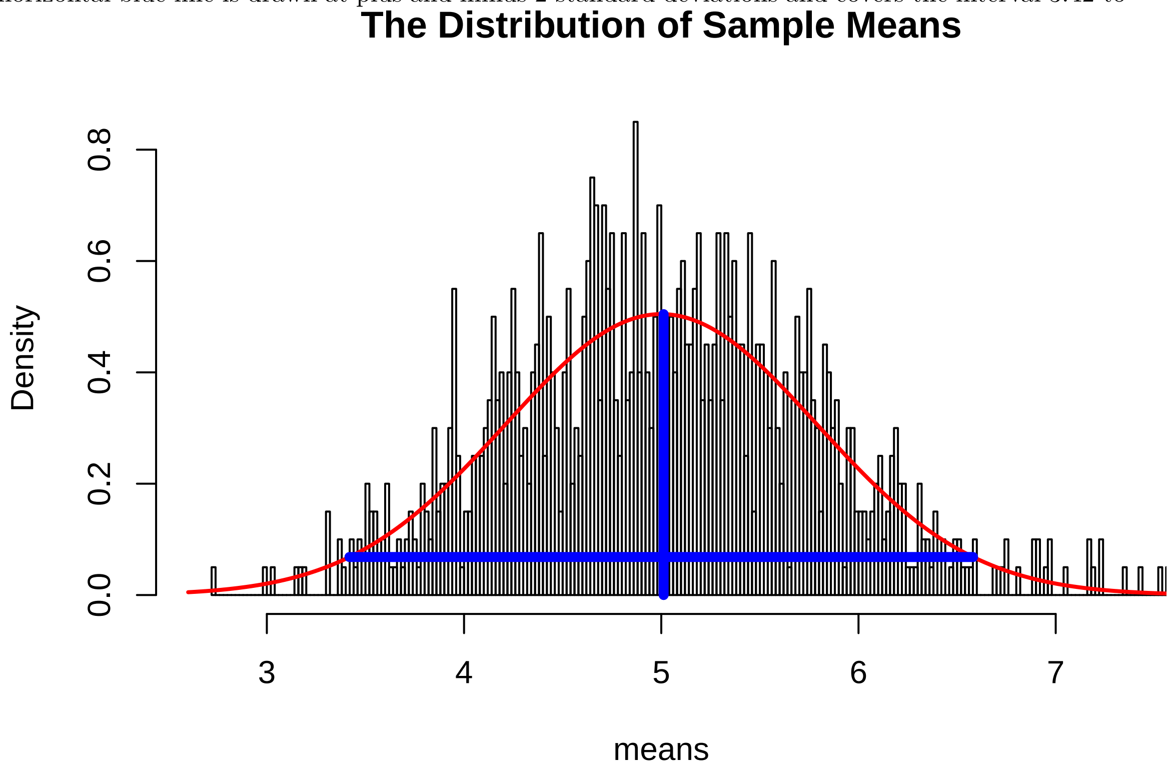


Figure 2 The Normallized Distribution of Sample Means

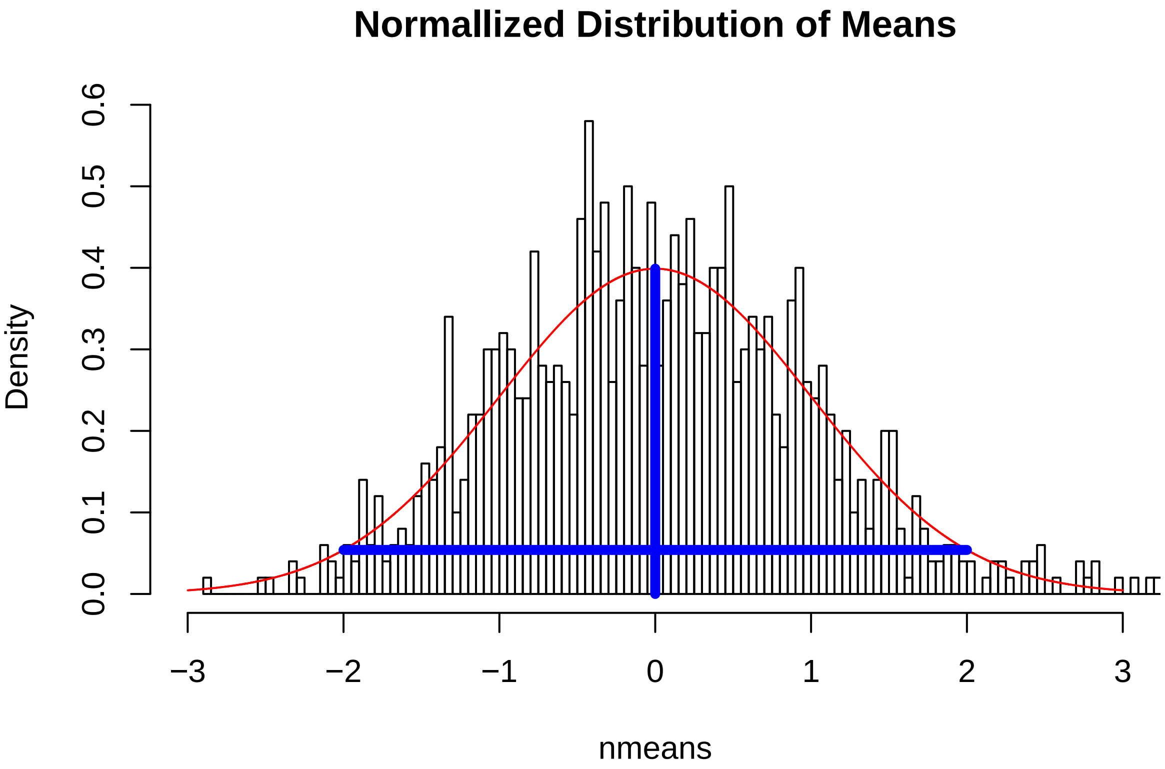
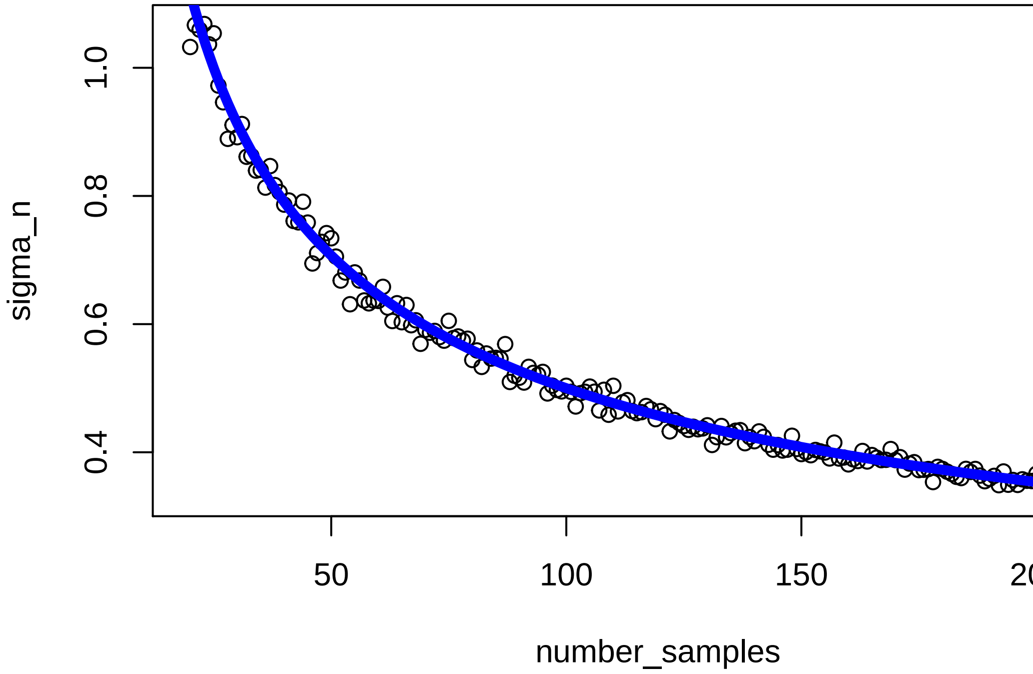


Figure 3 The Standard Deviation of the Mean by Number of Samples

- the blue line is sigma/sqrt(n) where sigma is the original population standard deviation, 1/lambda=5. and n is the number of samples


sigma_n
number_samples