

Lab 3a

Arjun Banik, Mary Lesperance

March 1, 2024

General Instructions

- Execute each chunk of code to ensure that your code works properly.
- Save this .Rmd file and then knit the entire document to pdf.

1. The Central Limit Theorem.

One of the most amazing results of probability is the Central Limit Theorem. The Central Limit Theorem states that sums of independent (identically distributed) random variables from almost any probability distribution have probability distributions that can be approximately calculated with areas under the normal distribution.

We demonstrate the theorem using simulations of sums of Uniform random variables.

```
# Define a function to generate <repeats> replications of an experiment.
# For each replication, simulate sample <size> Uniform(0, 1) random numbers.
# Sum the <size> random numbers for each replication and return a vector of length <repeats>.
# I code this efficiently without for loops using matrix computations instead of loops.
# For loops are computationally expensive, and in R we try to avoid them when we can.

Usim <- function(repeats, size){
  v <- runif(size * repeats) #generate size*repeats from Uniform(0,1)
  w <- matrix(v, nrow=size, ncol=repeats) #Put v in a matrix with size rows and repeats cols
  return(colSums(w)) #Sum the columns, the result has length repeats
}

set.seed(12345)
U1 <- Usim(10000, 1) #size 1
head(U1)

## [1] 0.7209039 0.8757732 0.7609823 0.8861246 0.4564810 0.1663718

U2 <- Usim(10000, 2) #size 2
head(U2)

## [1] 0.93372160 1.85087459 0.73360650 0.03704234 0.47392017 0.47249167

U5 <- Usim(10000, 5) #size 5
head(U5)

## [1] 2.179916 3.164076 2.075394 3.073965 2.691109 2.716220

U20 <- Usim(10000, 20) #size 20
head(U20) # sums of 20 U(0,1) random observations
```

```
## [1] 10.93786 10.23224 11.41482 10.48596 11.07793 10.20115
```

```
length(U20)  # check the length, should be 10000
```

```
## [1] 10000
```

```
# Draw histograms of the 4 vectors.
```

```
# look up the help file for the function layout
```

```
layout(matrix(c(1, 2, 3, 4), nrow=2, ncol=2, byrow=TRUE))
```

```
hist(U1, prob=TRUE, ann=FALSE)  #ann=FALSE turns off annotation
```

```
title("Sample Size 1")
```

```
hist(U2, prob=TRUE, ann=FALSE)
```

```
title("Sample Size 2")
```

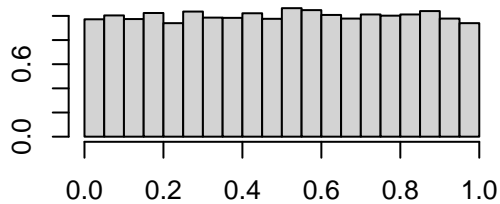
```
hist(U5, prob=TRUE, ann=FALSE)
```

```
title("Sample Size 5")
```

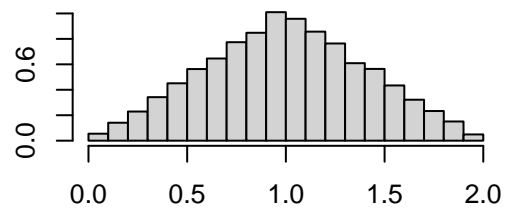
```
hist(U20, prob=TRUE, ann=FALSE)
```

```
title("Sample Size 20")
```

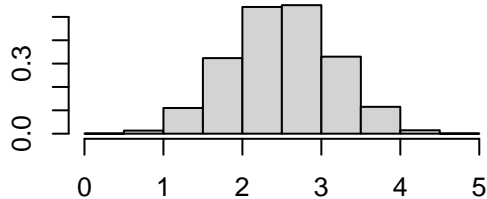
Sample Size 1



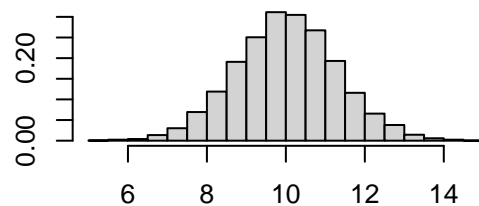
Sample Size 2



Sample Size 5



Sample Size 20



```
layout(1)
```

The successive histograms approach an approximate symmetric bell shape as the number (size) of random numbers in the sum gets larger.

2. The Normal(μ , σ^2) Likelihood graph.

We will plot the Likelihood surface as a function of μ and σ^2 for a set of data generated from the normal distribution.

```
#generate n Normal(mu=2, sig2=4) random observations

set.seed(12345)
n <- 100
x <- rnorm(n, mean=2, sd=2)

#Define the Likelihood function for vector mu and vector sig2
# given data vector x
# We loop over the values for mu and sig2 and create a matrix L
# L[i, j] contains the likelihood evaluated at mu[i] and sig2[j]

Lnorm <- function(mu, sig2, x){
  L <- matrix(0, nrow=length(mu), ncol=length(sig2))
  for(i in 1:length(mu)){
    for(j in 1:length(sig2)){
      L[i,j] <- prod(dnorm(x, mean=mu[i], sd=sqrt(sig2[j])))
    }
  }
  return(L)
}

muvals <- seq(0, 4, by=.05)
head(muvals)

## [1] 0.00 0.05 0.10 0.15 0.20 0.25

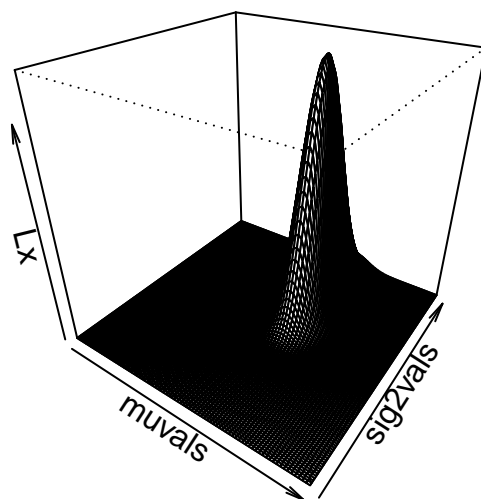
sig2vals <- seq(1, 7, by=.05)
head(sig2vals)

## [1] 1.00 1.05 1.10 1.15 1.20 1.25

Lx <- Lnorm(muvals, sig2vals, x)

persp(muvals, sig2vals, Lx, theta=35, phi=30) #3D plot
title("Normal Likelihood")
```

Normal Likelihood



```
# The MLE's of mu and sig2 are:
```

```
mean(x)
```

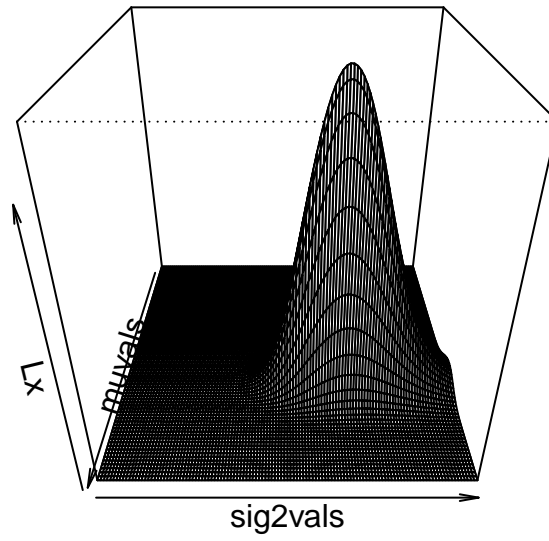
```
## [1] 2.490394
```

```
var(x)*(n-1)/n
```

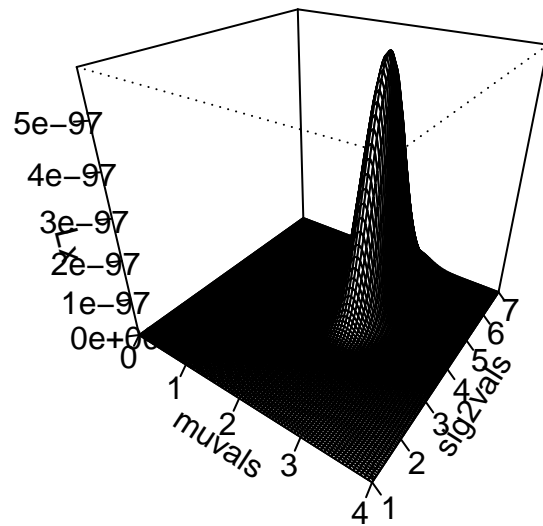
```
## [1] 4.920794
```

```
# Investigate the parameters theta and phi of the persp function, to see the  
# graph from different angles.
```

```
persp(muvals, sig2vals, Lx, theta=90, phi=30)
```



```
persp(muvals, sig2vals, Lx, theta=35, phi=30, ticktype="detailed")
```



The maximum of L_x should be at the Joint MLE's of μ and sig^2 !