CLT with R

Boyie Chen

11/15/2019

Central Limit Theorem

Recall: $X_i \sim (\mu, \sigma^2), i = 1, 2, 3, \ldots, n$ By CLT,

$$\frac{\bar{X_n} - \mu}{\sqrt{\frac{\sigma^2}{n}}} \rightarrow^d N(0,1)$$

as $n \rightarrow \infty$ where $\bar{X_n} \sim (\mu, \frac{\sigma^2}{n})$

Let's do the simulation and see how $\bar{X_n}$ behaves.

Generating pseudo data from known distributions

Main character : Sample Mean X__bar

Say $X_i \sim Binomial(n = 5, p = 0.1)$, $i = 1, 2, \ldots, 10000$ $E(X) = 5 \times 0.1 = 0.5$, $Var(X) = 5 \times 0.1 \times (1 - 0.1) = 0.45$

```
#X~Bin(n = 5, p = 0.1)
x1 = rbinom(10000, size = 5, prob = 0.1)
mean(x1)
```

```
## [1] 0.5048
```

```
var(x1)
```

```
## [1] 0.4612231
```

But this is what only one $\bar{X}$ behaves. If I want to see the how $\bar{X}$ distributed, I need a bunch of realizations of $\bar{X}$.

Writing a function to generate many realizations

I can simply repeat the process above and get many $\bar{X}$.

Define a function `x_bar_bin(n)` which helps me to get n realizations from $Bin(5, 0.1)$.

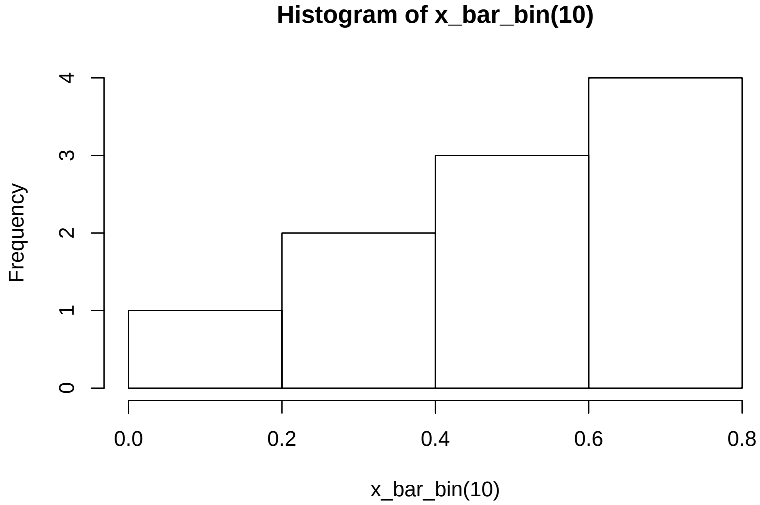
```
x_bar_bin = function(n){ #input: the number of realization of x_bar
  x_bar = c()
  for(i in 1:n){
    x1 = rbinom(n, size = 5, prob = 0.1)
    x_bar = rbind(x_bar, mean(x1))
  }
  return(x_bar)
}
```

When input is 10, then it gives me 10 realizations.

```
t(x_bar_bin(10)) #10 realization
```

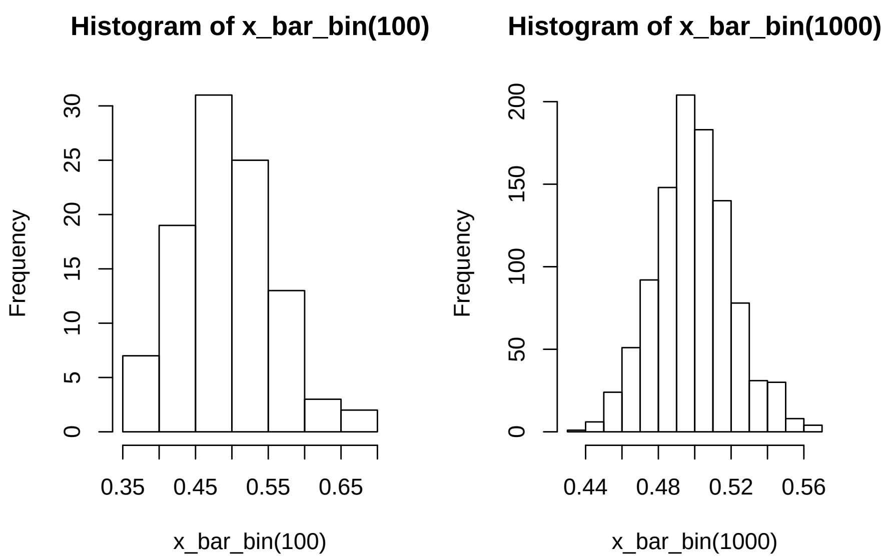
```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]  0.7  0.5  0.5  0.1  0.5  0.7  0.7  0.3  0.9   0.3
```

```r
hist(x_bar_bin(10)) #see the distribution of 10 realizations
```

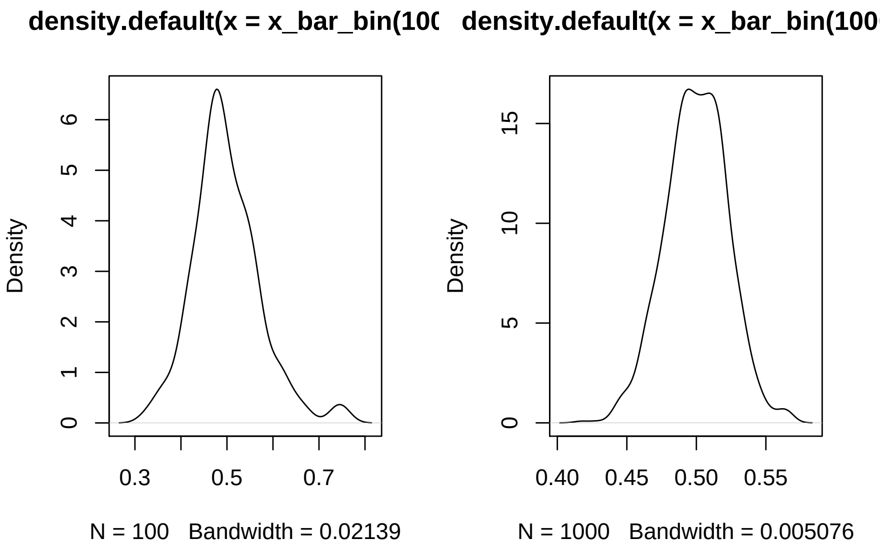


The followings are 100 realizations and 1000 realizations.

```r
par(mfrow = c(1,2))
hist(x_bar_bin(100))
hist(x_bar_bin(1000))
```




```r
#density(x_bar_bin(1000)) #check`density()` if you're interested in it
plot(density(x_bar_bin(100)))
plot(density(x_bar_bin(1000)))
```

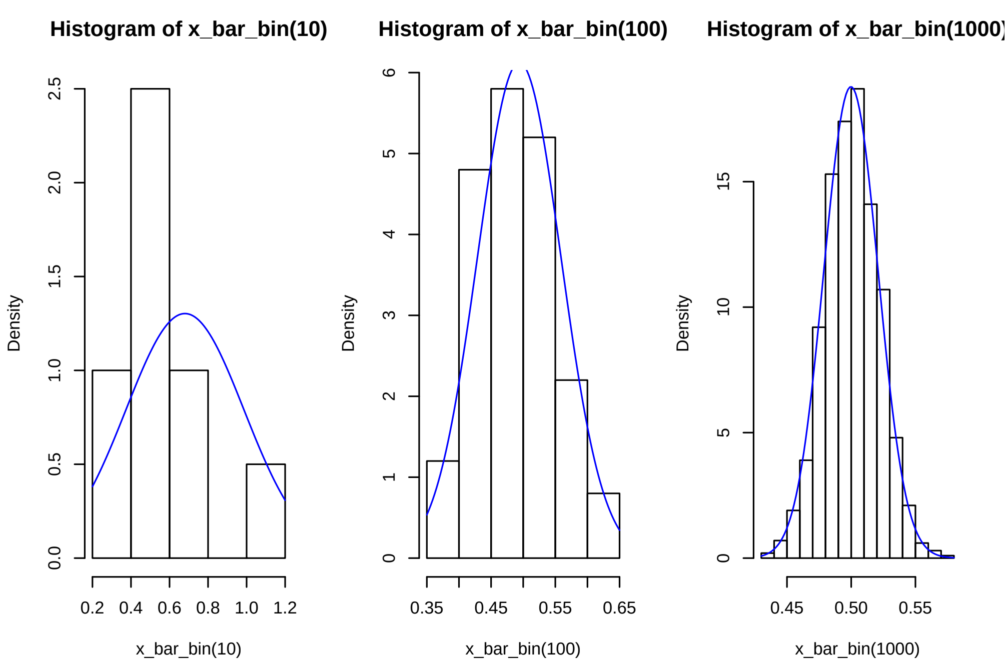



```
graphics.off()
```

Adding Normal Distribution Curve

We just define a function that helps us find $\bar{X}_n$ under different n. Let's add the normal curve and see how $\bar{X}_n$ fits normal distribution under different n.

```
set.seed(12345)
par(mfrow = c(1,3))
hist(x_bar_bin(10), freq = F)
curve(dnorm(x, mean = mean(x_bar_bin(10)), sd = sd(x_bar_bin(10))), add = T, col = 'blue')
hist(x_bar_bin(100), freq = F)
curve(dnorm(x, mean = mean(x_bar_bin(100)), sd = sd(x_bar_bin(100))), add = T, col = 'blue')
hist(x_bar_bin(1000), freq = F)
curve(dnorm(x, mean = mean(x_bar_bin(1000)), sd = sd(x_bar_bin(1000))), add = T, col = 'blue')
```



Normalization

We've seen how $\bar{X_n}$ behaves. Now let's see how normalized z behaves. Let

$$z = \frac{\bar{X_n} - \mu}{\sqrt{\frac{\sigma^2}{n}}}$$

We also define a function that returns the normalized realization of the r.v.

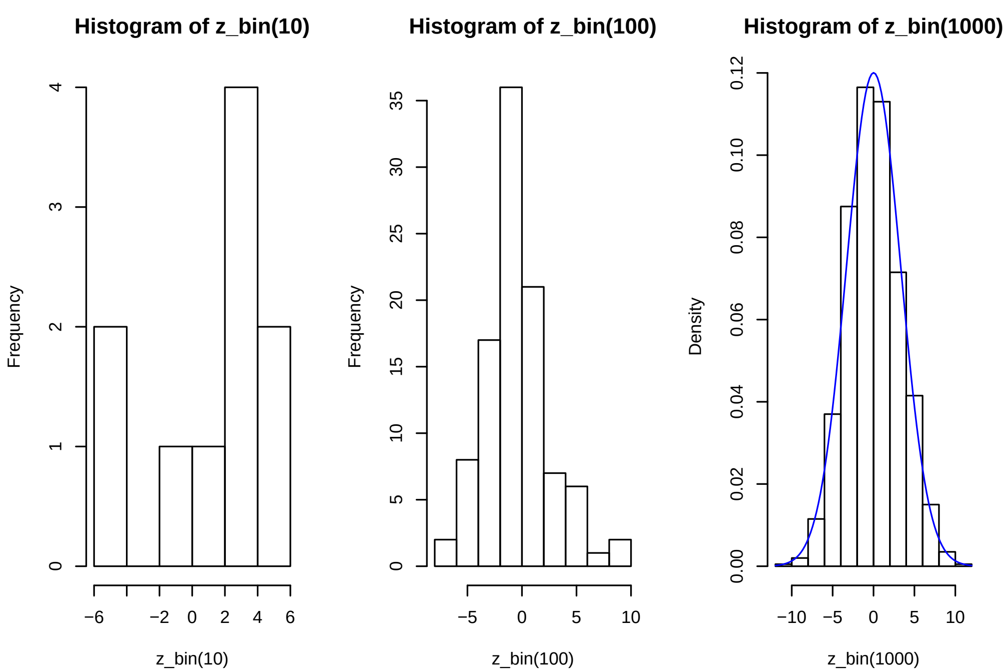
```
#See what the normalized r.v. behaves
#Z = sqrt(n)*(X_bar - mu)/sigma^2
z_bin = function(n){
  z = c()
  for(i in 1:n){
    x1 = rbinom(n, size = 5, prob = 0.1) #Bin(5, 0.1)
    #sqrt(n)*(mean(x1) - 5*0.1)/(5*0.1*0.9)^2 #normalization
    z = rbind(z, sqrt(n)*(mean(x1) - 5*0.1)/(5*0.1*0.9)^2)
  }
  return(z)
}
```

Similarly, `z_bin(n)` gives us n realizations of the normalized r.v.

```
t(z_bin(10)) #10 normalized realization
```

```
##           [,1]      [,2]      [,3]      [,4] [,5]     [,6]     [,7]
## [1,] -6.246474 -1.561619 -1.561619 -3.123237    0 7.808093 4.684856
##          [,8]     [,9]    [,10]
## [1,] 1.561619 4.684856 1.561619
```

```
par(mfrow = c(1,3))
hist(z_bin(10))
hist(z_bin(100))
hist(z_bin(1000), freq = F)
curve(dnorm(x, mean=mean(z_bin(1000)),sd=sd(z_bin(1000))),add=T, col="blue")
```



Large Matrix Approach

a more intuitive approach to look at the random realizations

```
head(rbinom(1000, size = 10, p = 0.1)) #generate 1000 random numbers from Bin(10,0.1)
```

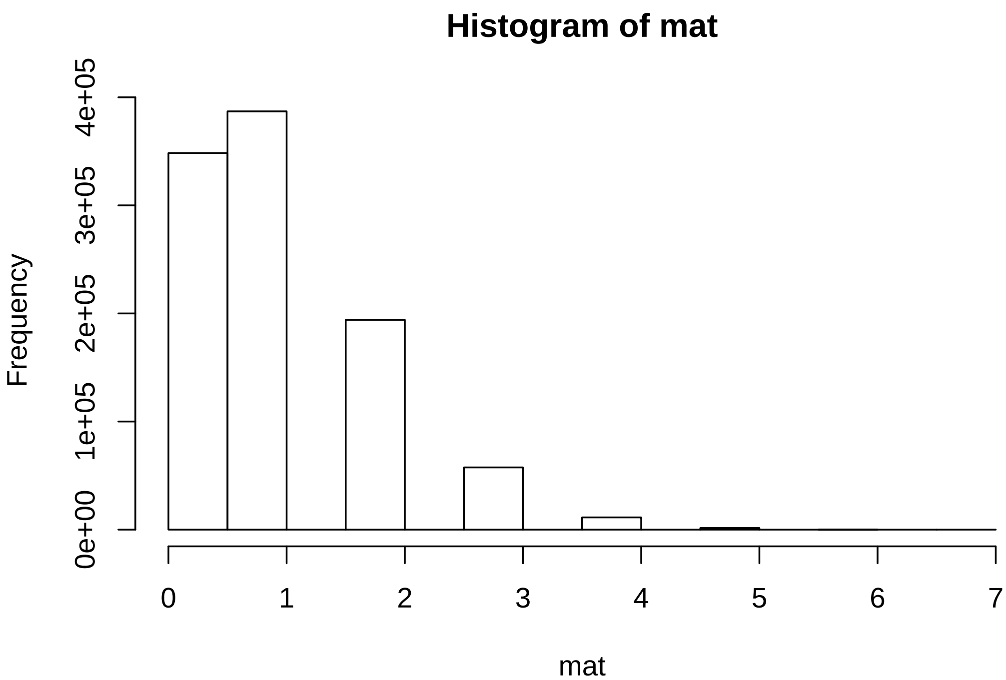
```
## [1] 1 1 0 0 0 3
```

```
mat = replicate(1000, rbinom(1000, size = 10, p = 0.1)) #generate 1000 sets of the above
dim(mat)
```

```
## [1] 1000 1000
```

If I just apply `hist()` on a matrix...

```
hist(mat) #overall dist. is Bin
```




```
#this is element-wise operation
```

The column of the matrix is one of 1000 sets of 1000 realizations. We can get 1000 realizations of $\bar{X}_n$ from 1000 columns.

```
#vertically sum up and divide by 1000 => get 1000 x_bars!
mean(mat[,1])
```

```
## [1] 1.052
```

```
mean(mat[,2])# on and on and on...
```

```
## [1] 0.98
```

With the for loop.

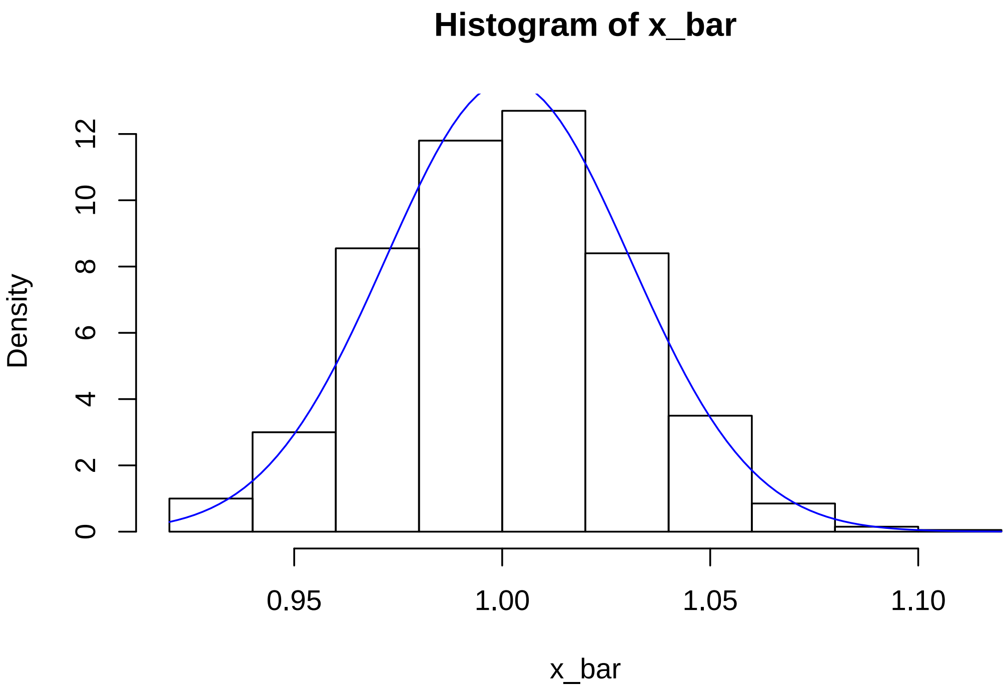
```
x_bar = c()
for(i in 1:1000){
  x_bar = rbind(x_bar, mean(mat[,i]))
}
t(head(x_bar))
```

```
##       [,1] [,2]  [,3]  [,4]  [,5]  [,6]
## [1,] 1.052 0.98 0.991 1.006 1.011 1.035
```

Density Plot

If we look at the "density" and compare it to normal density.

```
hist(x_bar, freq = F)
curve(expr = dnorm(x, mean = mean(x_bar), sd = sd(x_bar)), add = T, col = 'blue')
```



Alternative Expression (Optional)

We can also draw the "Frequency Plot" instead of density plot. But we have to adjust the "height" of normal curve.

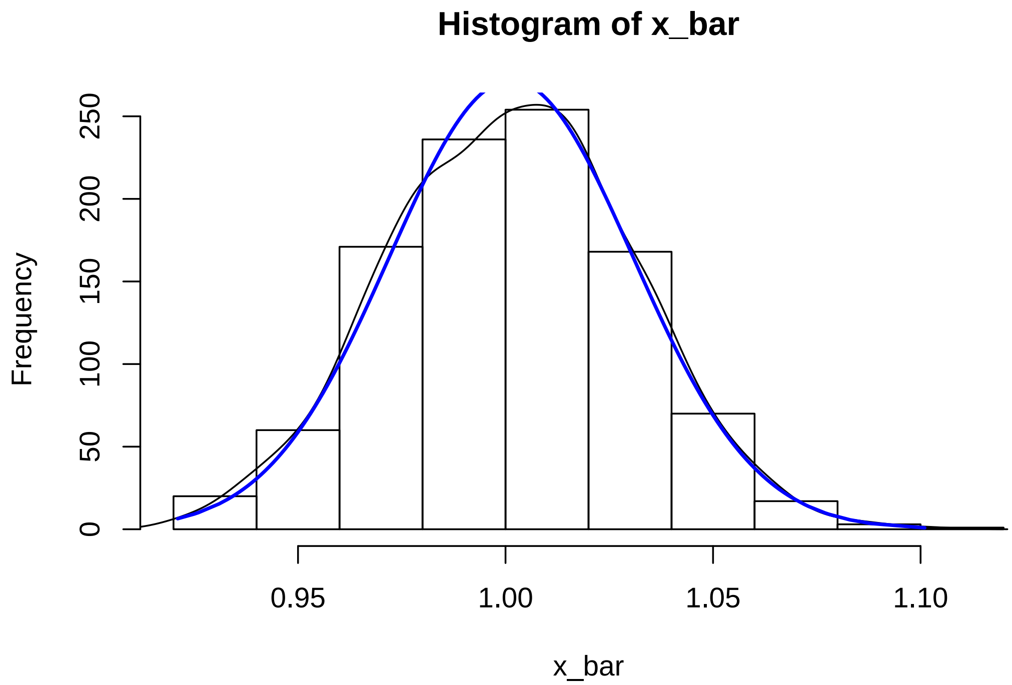
```
x_bar.hist = hist(x_bar)
multiplier = x_bar.hist$counts/x_bar.hist$density

multiplier[1]
```

```
## [1] 20
```

```
x_bar.den = density(x_bar)
x_bar.den$y = x_bar.den$y*multiplier[1]

#If we look ar the "frequency"...compare to normal
hist(x_bar)
lines(x_bar.den)
lines(sort(x_bar), dnorm(x = sort(x_bar), mean = mean(x_bar), sd = sqrt(var(x_bar))) * multiplier[1], c
```



Also, for the normalized r.v., it should fits normal curve.

```
#normalization
z = (x_bar - 10*0.1)/(sqrt(10*0.1*0.9)/sqrt(1000))
h = hist(z, freq = F)
curve(expr = dnorm(x, mean = mean(z), sd = sd(z)), add = T, col = 'red')
```