Homework #3

1. (5 points each) Exercise 8.5.4, 8.5.15, 9.5.1

2. (10 points each) Exercise 8.5.14, 9.5.8, 9.5.9, 9.5.13 (The data set is in the file lowbwt.txt in the directory ASCII\exercise of the CD in the textbook)

3. (10 points) Run the simulation program for CLT in the following page. Try several types of random variables: Binomial(6,0.03), Poisson(2) and Uniform(0,5). At what sample size does the distribution of means in your simulation come close to normal distribution?

4. (5 points) Generate independent random variable X and Y, respectively, from N(0,1) and a F-distribution with degrees of freedoms 4 and 9. Use nobs=4000 simulated observations of each variable to approximate the mean **$E(X^2/Y)$**. Submit the R codes with your answer.

==

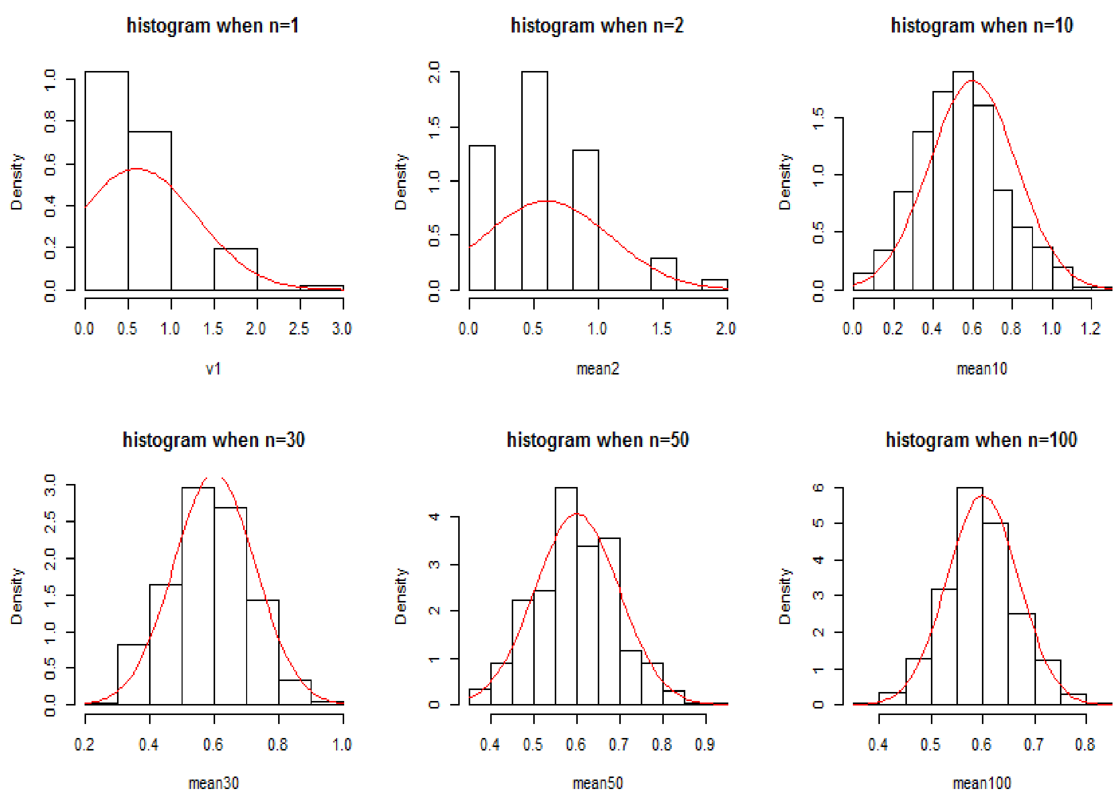
R example for illustrating CLT

```
### A simulated example illustrating CLT
## Generate n=100 random variables from a Binomial(3, 0.2) distribution
##  Have n.obs=400 observations of each variable.
n <- 100
n.obs <- 400
rand.data <- matrix(rbinom(n.obs*n, size=3, prob=0.2), nrow=n.obs)

## Create summary variables;
# v1--the first variable, mean2 -- average of the first two variables
# mean10 -- average of the first 10 variables, similarly mean30, etc.
v1 <- rand.data[,1] #The first column (variable)
#Average of the first two columns(variables),
# apply function 'mean' on margin 1 (row-wise function application)
mean2 <- apply(rand.data[,1:2], FUN=mean, MARGIN=1)
#Average of the first ten columns(variables)
mean10<- apply(rand.data[,1:10], FUN=mean, MARGIN=1)
#Average of the first 30, 50, 100 columns (variables)
mean30 <- apply(rand.data[,1:30], FUN=mean, MARGIN=1)
mean50 <- apply(rand.data[,1:50], FUN=mean, MARGIN=1)
mean100 <- apply(rand.data[,1:100], FUN=mean, MARGIN=1)

##Plot the histograms of the summary variables above, overlay with
#   the density plot of the normal distribution in the CLT.
mu    <- 3*0.2                #Binomial mean
sigma <- sqrt(3*0.2*(1-0.2))  #Binomial standard deviation
## Arrange 6 plots in one page 2 by 3
par(mfrow=c(2,3))
# Histogram of v1,
#   then add the normal density curve in red color (col=2)
hist(v1, freq = FALSE, main="histogram when n=1")
curve(dnorm(x, mean=mu, sd=sigma),col=2,add=T)
# Histograms of mean2, mean10, ....
hist(mean2, freq = FALSE, main="histogram when n=2")
curve(dnorm(x, mean=mu, sd=sigma/sqrt(2)),col=2,add=T)
hist(mean10, freq = FALSE, main="histogram when n=10")
curve(dnorm(x, mean=mu, sd=sigma/sqrt(10)),col=2,add=T)
```

```
hist(mean30, freq = FALSE, main="histogram when n=30")
curve(dnorm(x, mean=mu, sd=sigma/sqrt(30)),col=2,add=T)
hist(mean50, freq = FALSE, main="histogram when n=50")
curve(dnorm(x, mean=mu, sd=sigma/sqrt(50)),col=2,add=T)
hist(mean100, freq = FALSE, main="histogram when n=100")
curve(dnorm(x, mean=mu, sd=sigma/sqrt(100)),col=2,add=T)
```

The above program produces 100 columns of random number from Binomial (3, 0.2) distribution and also calculate means of the first 2, 10, 30, 50, 100 columns. Then histograms of n.obs=400 repeated simulations were then drawn. Run this program and see how the shapes of histogram change.

To generate other types of random variables:
Poisson(lambda): rpois(, lambda=);
Uniform(0,1): runif( );
Normal(0,1): rnorm( );



There is a R applet at
https://adamding.shinyapps.io/CLTadamding/
You may try that one out with more examples. The underlying code is similar to the example here.