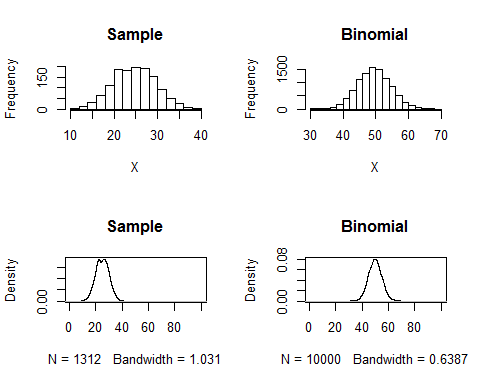
Assessment1-ST5222

Name: Zhu Xu ; User ID: E0337988 ; Student ID: A0191344H

15 September 2018

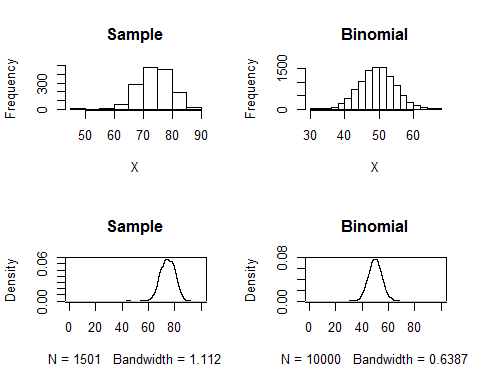
## Proposal–Unif , Binomial distribution(n=100, p=0.25)

f\_0 <- function(x){  
 (factorial(100)/(factorial(x)\*factorial(100-x)))\*0.25^(x)\*(1-0.25)^(100-x)  
 }   
# define the Binomial distribution , n=100, p=0.25  
q\_0 <- function(x){1/100}   
# define the proposal function q\_0 (Uniform)  
n <- 100   
X <- numeric() # store samples  
count <- 1   
x\_old <- 0 # initialize the Markov chain  
for(i in 1:10000) # main loop to obtain samples  
{  
 x\_new <- sample.int(101, 1)-1 # sample a integer from q\_0, as a candidate   
 u <- runif(1,0,1)   
 acc\_1 <- min((f\_0(x\_new)\*q\_0(x\_old))/(f\_0(x\_old)\*q\_0(x\_new)), 1)  
# calculate the acceptance prob  
 if(u < acc\_1) # accept  
 {  
 X[count] <- x\_new  
 count <- count +1  
 x\_old <- x\_new  
 }  
 else # reject the sample  
 {  
 x\_old <- x\_old  
 }  
}  
par(mfrow = c(2, 2))  
hist(X, main = "Sample", xlab = "X")  
XX <- rbinom(10000, 100, 0.5)   
hist(XX, main = "Binomial", xlab = "X")  
P <- density(X, n = 10000, from = 1, to = 100)  
plot(P, main = "Sample")  
B <- density(XX, n = 10000, from = 1, to = 100)  
plot(B, main = "Binomial")



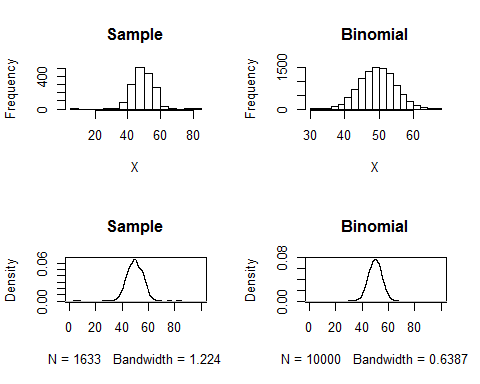
## proposal–Unif , Binomial distribution(n=100, p=0.75)

f\_2 <- function(x){  
 (factorial(100)/(factorial(x)\*factorial(100-x)))\*0.75^(x)\*(1-0.75)^(100-x)  
 }   
# define the Binomial distribution , n=100, p=0.75  
q\_0 <- function(x){1/100}  
# define the proposal function q\_0 (Uniform)  
n <- 100   
X <- numeric() # store samples  
count <- 1   
x\_old <- 0 # initialize the Markov chain  
for(i in 1:10000)  
{  
 x\_new <- sample.int(101, 1)-1 # sample a integer from q\_0, as a candidate   
 u <- runif(1,0,1)   
 acc\_1 <- min((f\_2(x\_new)\*q\_0(x\_old))/(f\_2(x\_old)\*q\_0(x\_new)), 1)   
# calculate the acceptance prob  
 if(u < acc\_1) # accept  
 {  
 X[count] <- x\_new  
 count <- count +1  
 x\_old <- x\_new  
 }  
 else # reject  
 {  
 x\_old <- x\_old  
 }  
}  
par(mfrow = c(2, 2))  
hist(X, main = "Sample", xlab = "X")  
XX <- rbinom(10000, 100, 0.5)   
hist(XX, main = "Binomial", xlab = "X")  
P <- density(X, n = 10000, from = 1, to = 100)  
plot(P, main = "Sample")  
B <- density(XX, n = 10000, from = 1, to = 100)  
plot(B, main = "Binomial")

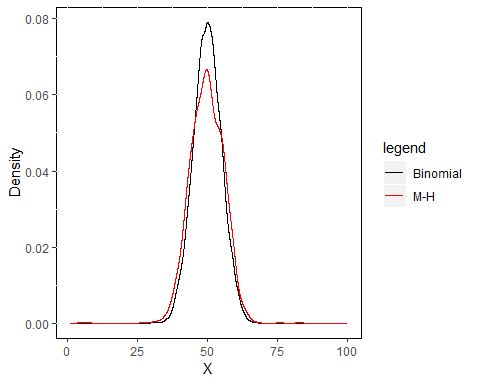


## proposal — unif 3, p=0.5

f\_1 <- function(x){  
 (factorial(100)/(factorial(x)\*factorial(100-x)))\*0.5^(x)\*(1-0.5)^(100-x)  
 }   
# define the Binomial distribution , n=100, p=0.5  
q\_0 <- function(x){1/100}   
# define the proposal function q\_0 (Uniform)  
n <- 100   
X <- numeric() # store samples  
count <- 1   
x\_old <- 0 # initialize the Markov chain  
for(i in 1:10000)  
{  
 x\_new <- sample.int(101, 1)-1 # sample a integer from q\_0, as a candidate   
 u <- runif(1,0,1)  
 acc\_1 <- min((f\_1(x\_new)\*q\_0(x\_old))/(f\_1(x\_old)\*q\_0(x\_new)), 1)   
# calculate the acceptance prob  
 if(u < acc\_1) # accept  
 {  
 X[count] <- x\_new  
 count <- count +1  
 x\_old <- x\_new  
 }  
 else # reject  
 {  
 x\_old <- x\_old  
 }  
}  
par(mfrow = c(2, 2))  
hist(X, main = "Sample", xlab = "X")  
XX <- rbinom(10000, 100, 0.5)   
hist(XX, main = "Binomial", xlab = "X")  
P <- density(X, n = 10000, from = 1, to = 100)  
plot(P, main = "Sample")  
B <- density(XX, n = 10000, from = 1, to = 100)  
plot(B, main = "Binomial")  
  
p <- 0.5  
real <- rbinom(10000, 100, p)  
d <- density(real, n = 10000, from = 1, to = 100)  
s <- density(X, n = 10000, from = 1 ,to = 100)  
Data\_real <- data.frame(num = d$x, y = d$y, legend = rep(c("Binomial "), 10000, 1))  
Data\_sample <- data.frame(num = s$x, y = s$y, legend = rep(c("M-H"), 10000, 1))  
Data <- rbind(Data\_real, Data\_sample)  
library(ggplot2)

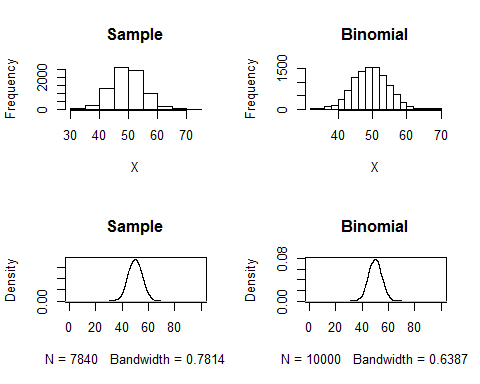


g <- ggplot( Data, aes(x = num, y = y, group = legend) ) +   
geom\_line( aes(color = legend), size = 0.5 ) +  
xlab("X") + ylab("Density") +   
scale\_color\_manual(values = c("black", "red")) +   
theme(panel.background = element\_rect(fill = 'white', colour = 'black'))  
g



## proposal — poisson lamdba=50

lambda <- 50  
f\_1 <- function(x){  
 (factorial(100)/(factorial(x)\*factorial(100-x)))\*0.5^(x)\*(1-0.5)^(100-x)  
 }   
# define the Binomial distribution , n=100, p=0.5  
q\_1 <- function(x){((lambda^x)\*exp(-lambda))/factorial(x)}   
# define the proposal function q\_1 (Poisson), lambda=np=50  
n <- 100   
X <- numeric() # store samples  
count <- 1   
x\_old <- 0 # initialize the Markov chain  
for(i in 1:10000)  
{  
   
 x\_new <- rpois(1, lambda) # sample a candidate from q\_1   
 u <- runif(1,0,1)  
 acc\_1 <- min((f\_1(x\_new)\*q\_1(x\_old))/(f\_1(x\_old)\*q\_1(x\_new)), 1)   
# calculate the acceptance prob  
 if(u < acc\_1) # accept  
 {  
 X[count] <- x\_new  
 count <- count +1  
 x\_old <- x\_new  
 }  
 else # reject  
 {  
 x\_old <- x\_old  
 }  
}  
par(mfrow = c(2, 2))  
hist(X, main = "Sample", bty = "o", xlab = "X")  
XX <- rbinom(10000, 100, 0.5)   
hist(XX, main = "Binomial", xlab = "X")  
P <- density(X, n = 10000, from = 1, to = 100)  
plot(P, main = "Sample")  
B <- density(XX, n = 10000, from = 1, to = 100)  
plot(B, main = "Binomial")



p <- 0.5  
real <- rbinom(10000, 100, p)  
d <- density(real, n = 10000, from = 1, to = 100)  
s <- density(X, n = 10000, from = 1 ,to = 100)  
Data\_real <- data.frame(num = d$x, y = d$y, legend = rep(c("Binomial "), 10000, 1))  
Data\_sample <- data.frame(num = s$x, y = s$y, legend = rep(c("M-H Sample"), 10000, 1))  
Data <- rbind(Data\_real, Data\_sample)  
library(ggplot2)  
g <- ggplot( Data, aes(x = num, y = y, group = legend) ) +   
geom\_line( aes(color = legend), size = 0.5 ) +  
xlab("X") + ylab("Density") +   
scale\_color\_manual(values = c("black", "red")) +   
theme(panel.background = element\_rect(fill = 'white', colour = 'black'))  
g

