#Tyler Sulsenti  
#HW 2  
#1.  
  
#i  
p = 0.4  
m = 8.25  
n <- c(20,30,50,100)  
for(i in 1:4) {  
 print(paste("P(N<=8.25) for n =", n[i] ,"is:", pbinom(8.25, n[i], p)))  
}

## [1] "P(N<=8.25) for n = 20 is: 0.595598725312224"  
## [1] "P(N<=8.25) for n = 30 is: 0.0940112158300952"  
## [1] "P(N<=8.25) for n = 50 is: 0.000230522860591324"  
## [1] "P(N<=8.25) for n = 100 is: 5.43112664040676e-13"

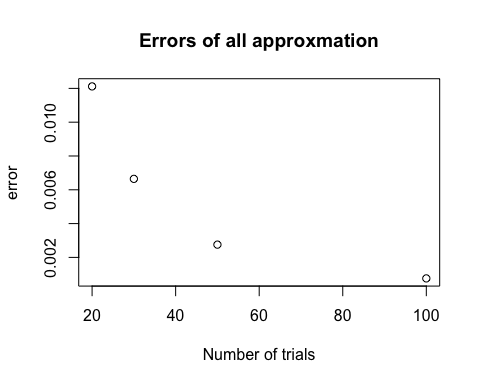
#ii  
## Laplace Theorem  
for(i in 1:4){  
 num = (m - n[i]\*p)  
 sigma = sqrt((n[i]\*p)\*(1-p))  
 print(paste("The normal approximation through the Laplace Theorem for n=", n[i], "is:", pnorm(num/sigma)))  
}

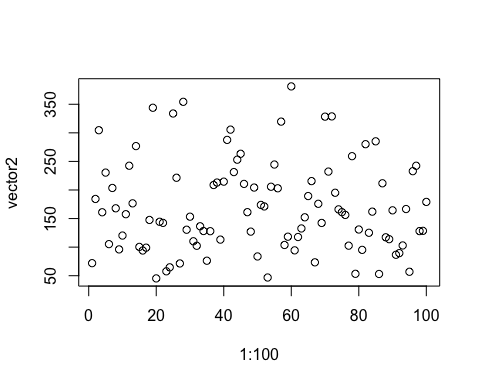
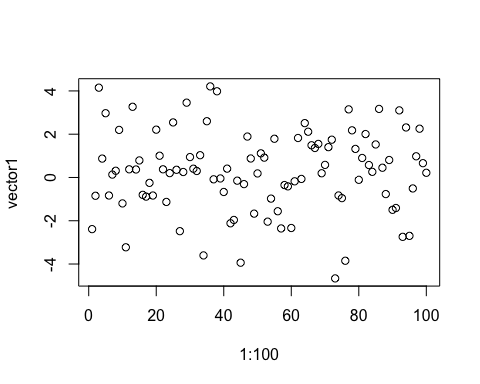
## [1] "The normal approximation through the Laplace Theorem for n= 20 is: 0.545424253017067"  
## [1] "The normal approximation through the Laplace Theorem for n= 30 is: 0.0811252499236159"  
## [1] "The normal approximation through the Laplace Theorem for n= 50 is: 0.000347007256425347"  
## [1] "The normal approximation through the Laplace Theorem for n= 100 is: 4.55759723740118e-11"

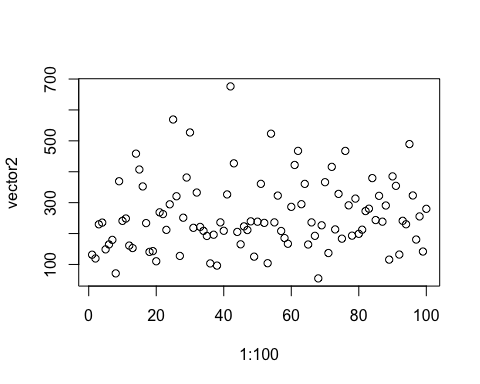
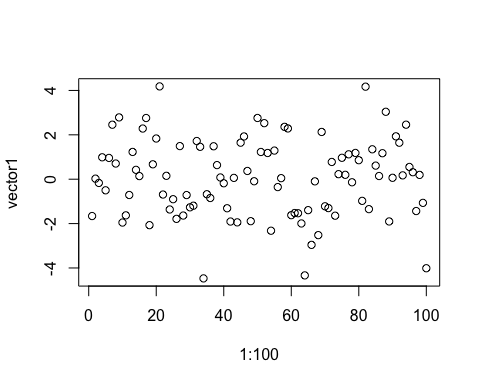
#iii  
error = c()  
for(i in 1:4){  
 phat = 8.25/n[i]  
 errorApprox = (phat\*(1-phat))/n[i]  
 error[i] = errorApprox  
 print(paste("The error of approxmination for n=", n[i], "is:", error[i]))  
}

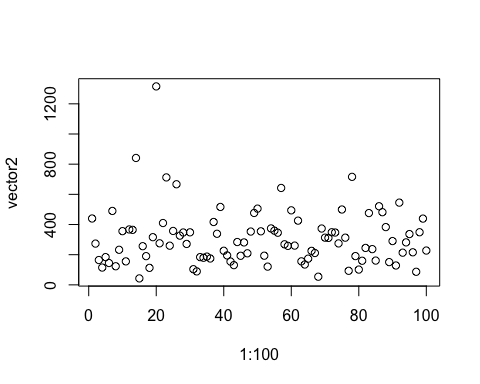
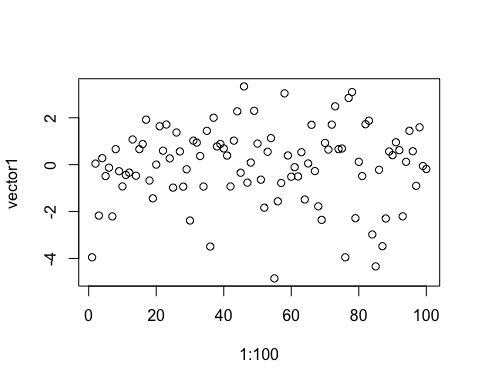
## [1] "The error of approxmination for n= 20 is: 0.0121171875"  
## [1] "The error of approxmination for n= 30 is: 0.00664583333333333"  
## [1] "The error of approxmination for n= 50 is: 0.0027555"  
## [1] "The error of approxmination for n= 100 is: 0.0007569375"

plot(n,error, xlab ="Number of trials")  
title(main = "Errors of all approxmation")



#iv  
## Based on the error plot in iii, i can see that the error gets smaller and smaller the more trials, n, that there are.  
#2 -> this function covers the compuations for i, ii, and iii.  
hw2\_func = function(n, times){  
 vector1 = c()  
 vector2 = c()  
 for(i in 1:times){  
 x = rnorm(n,2,3)  
 xbar = mean(x)  
 numerator = xbar-2  
 denominator = sqrt((3) / n)  
 compute1 = numerator/denominator  
 numerator = (n-1) \* var(x)^2   
 compute2 = numerator/3^2   
 vector1 = c(vector1, compute1)  
 vector2 = c(vector2, compute2)  
 }  
 plot(1:100, vector1)  
 plot(1:100, vector2)  
}  
#i hw2\_func(20, 100) 

#ii hw2\_func(30, 100) 

#iii hw2\_func(50, 100) 

#iv  
## We can see based upon the information in i, ii, and iii, that while n increases in value, the result of the fucntion (xbar - 2)/sqrt((3^2) / n) stays near 0 or becomes 0  
## However, in the fucntion ((n - 1)S^2)/3^2, as the value of n increases, so does the reuslt of the function