Problem Set 3

Antonio Dehesa

# Introduction

These questions were rendered in R markdown through RStudio (<https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf>, <http://rmarkdown.rstudio.com> ).

Please generate your solutions in R markdown, without later editing, and upload both a knitted doc, docx, or pdf document in addition to the Rmd file. Please be sure that the knitted document displays the results of your calculations.

The questions in this problem set use material from the slides on parameter estimation.

# Question 1

## Context

We have seen that count of successes from Bernoulli trials is modeled as then the maximum likelihood estimate of equals . Suppose this is repeated times and successes are observed in Bernoulli trials, successes are observed in Bernoulli trials, and so on through successes in Bernoulli trials. The goal to find the maximum likelihood estimate of if these are modeled as samples from , , and so on through .

## Q1, part 1

(5 points)

Consider the likelihood function for as outcomes from independent binomial distributions .

1. Please give the likelihood function .

M = Number of trials n = Each of the trials, one by one, from 1 to M As these are independent distributions, the solution would be the product of each individual distribution

B) Please give the log of the likelihood function as a sum of terms of the form ## Q1, part 2

(5 points)

1. Please give the derivative with respect to of . (your answer here)

If we consider as a constant, then the derivative would be 0. As we can see, it has no p terms, therefore, its a constant. d/dp of ki*log(p) = 1/p*ki = ki/p d/dp of (n\_i-k\_i)log(1-p) = We can consider n\_i-k\_i as constants, as they do not depend on p. Therefore, the derivative should be: -(n\_i-k\_i)/(1-p)

Therefore, the final result should be:

1. Please give the value of that maximizes . (your answer here)

We get this result by using the property that sum of (a-b) = sum of a - sum of b to split sum of (k\_i - n\_i). With this, we can then get: which would eventually lead us to the final result: ## Q1, part 3

(5 points)

If the samples from independent binomial distributions are viewed as successes in independent Bernoulli trials with probability of success equal to , what is the maximum likelihood estimate of . (your answer here) Basically, P: This P is already maximized, as it is the result we got from the previous question. In summary, P = sum of k / sum of n, which is basically the same as k/n, which is the same as the one obtained in the slides. # Question 2

## Context

The code below generates a sample, samp1, of size 10,000 from the distribution and a sample, samp2 of size 10,000 from the distribution.

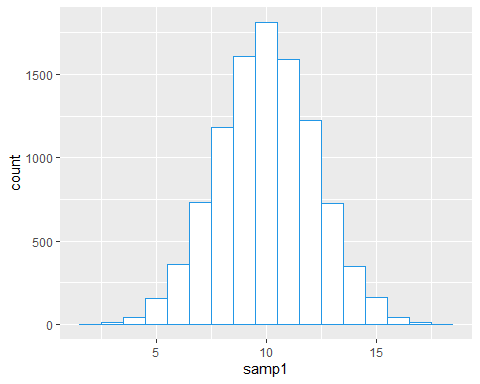
set.seed(12345)  
samp1<-rbinom(10000,20,.5)  
dat1<-data.frame(x=samp1)  
samp2<-rbinom(10000,50,.3)  
dat2<-data.frame(x=samp2)

## Q2, part 1

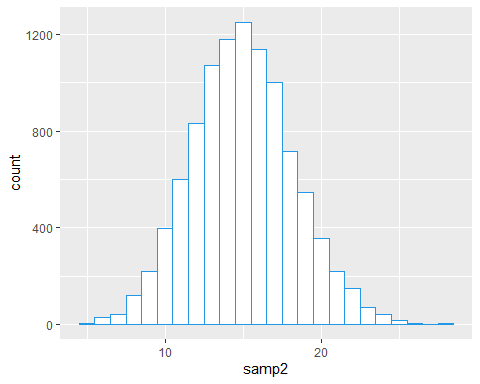
(5 points)

Please display separate histograms of samp1 and samp2 with binwidth equal to 1. (your code and plots here)

ggplot(dat1, aes(x = samp1)) +   
 geom\_histogram(colour = 4, fill = "white",   
 binwidth = 1)



ggplot(dat2, aes(x = samp2)) +   
 geom\_histogram(colour = 4, fill = "white",   
 binwidth = 1)



## Q2, part 2

(5 points)

Treating samp1 and samp2 as samples from Normal distributions and , please give maximum likelihood estimates of , , , and . (your answer here)

## Answer Q2P2

mu\_1 = mean(samp1)  
mu\_2 = mean(samp2)  
  
sigma\_1 = sd(samp1)  
sigma\_2 = sd(samp2)  
  
mu\_1

## [1] 10.0043

mu\_2

## [1] 14.9743

sigma\_1

## [1] 2.222786

sigma\_2

## [1] 3.252276

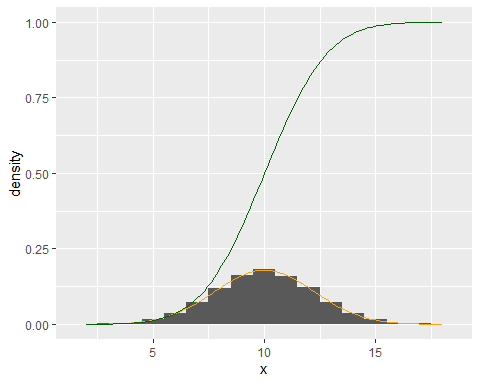
## Q2, part 3

(5 points)

The plotting methods from continuous\_probability\_distributions\_2\_4\_2.Rmd and practice problem set 2 may be useful here.

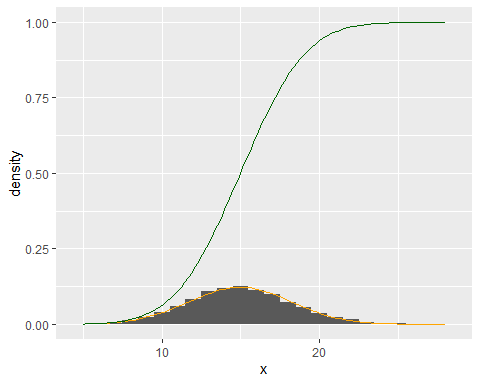
1. For samp1 please display the density histogram with density curve for superimposed. (your code and plot here)

plot\_1<- ggplot(dat1, aes(x=x)) +  
 geom\_histogram(aes(y = after\_stat(density)),binwidth=1) +  
 stat\_function(fun = dnorm, colour = "orange",args = list(mean =mu\_1, sd = sigma\_1))+  
 stat\_function(fun = pnorm, colour = "darkgreen",args = list(mean =mu\_1, sd = sigma\_1))  
plot\_1



1. For samp2 please display the density histogram with density curve for superimposed. (your code and plot here)

plot\_2<- ggplot(dat2, aes(x=x)) +  
 geom\_histogram(aes(y = after\_stat(density)),binwidth=1) +  
 stat\_function(fun = dnorm, colour = "orange",args = list(mean =mu\_2, sd = sigma\_2))+  
 stat\_function(fun = pnorm, colour = "darkgreen",args = list(mean =mu\_2, sd = sigma\_2))  
plot\_2



# Question 3

Please carry out the analysis below and answer the questions that follow. For this assignment, please do all calculations in R and show the code and the results in the knit document. Some calculations that may be useful are shown. You are not required to use these.

## Context

In statistical analyses, there are some distributions that aren’t Normal, but that can be well-approximated by Normal distributions. From the plots above, some binomial models appear to be good candidates.

To examine this, we will view plots of binomial densities and corresponding Normal distributions for a collection of binomial distributions. We will also compute a measure of the error of approximating the binomial distribution by the corresponding Normal distribution.

Above, we found an appropriate Normal distribution for a given binomial distribution by using maximum likelihood estimates of and based on a sample from a binomial distribution. Here, we calculate the values of and corresponding to the given binomial distribution under the assumption that the proportion of each outcome exactly equals that given by the density function for the binomial distribution.

Suppose the binomial distribution is . The maximum likelihood estimate for based on a sample is replaced by where is the density function for . To see that this is reasonable, rewrite as . For large , the value will occur approximately times in the sample , so the term is approximately .

Similarly, The maximum likelihood estimate for based on a sample is replaced by .

The definition of a function to find the values of the parameters “mean” and “sd” for the Normal distribution corresponding to a binomial distribution with parameters “size”=“sz” and “prob”=“p” follows:

normal.parameters.get<-function(sz,p){  
 mu<-sum(0:sz\*dbinom(0:sz,sz,p))# weighted average of the binomial outcomes  
 sigma.sq<-sum((0:sz-mu)^2\*dbinom(0:sz,sz,p))  
 return(c(mu,sqrt(sigma.sq)))  
}

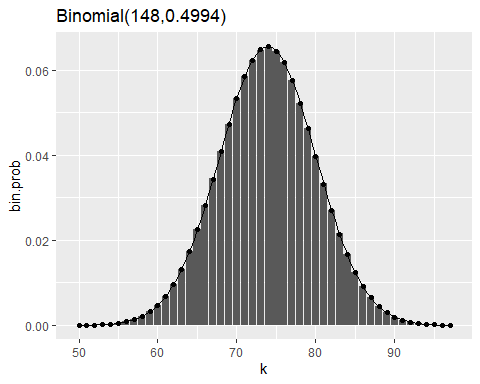
The measure of the error used is the sum of the squared differences between the probability of an outcome under the binomial distribution and the probability of a value in under the corresponding Normal distribution.

The definition follows of a function to find this sum of squared differences in probability for a binomial distribution with size parameter equal to “sz” and probability parameter equal to “pr” and the probabilities according to the corresponding Normal distribution:

# Function to calculate a sum of square errors is estimating  
# a binomial distribution with parameters sz and pr  
# by a Normal distribution with parameters norm=(mean,sd) as computed by   
# normal.parameters.get  
  
q1.approx<-function(sz.this,pr.this){  
 normal.params<-normal.parameters.get(sz.this,pr.this)  
 m<- normal.params[1] # mu  
 s<-normal.params[2] # sigma  
 # 0.5 below the binomial outcomes {0,1,...sz}  
 lower<-0:sz.this-.5  
 # 0.5 above the binomial outcomes {0,1,...sz}  
 upper<-0:sz.this+.5  
 # Probabilities under the Normal distribution  
 # for the events [k-.5,k+.5] for k in {0,1,...sz}   
 normal.probs<-pnorm(upper,m,s)-pnorm(lower,m,s)  
 # Sum of the squares of the differences between   
 # the probability of the event [k-.5,k+.5] under the Normal   
 # distribution and the probability of k under the binomial  
 # distribution  
 error<-sum((normal.probs-dbinom(0:sz.this,sz.this,pr.this))^2)  
 return(error)  
}

For example, apply these to the performance of the Normal approximation for . These parameters were chosen based on the number of paralytic polio cases and the proportion of vaccinated children from the randomized control trial in the polio data.

data("PolioTrials")  
dat<-PolioTrials  
sz<-sum(dat$Paralytic[1:2])  
prop<-dat$Population[1]/sum(dat$Population[1:2])  
normal.params<-normal.parameters.get(sz,prop)  
mu<-normal.params[1]  
sigma<-normal.params[2]  
  
# Plot only values of k that are at all likely.  
k<-qbinom(.00005,sz,prop):qbinom(.99995,sz,prop)  
  
# Make a data frame of these values of k, their probability under the binomial distribution, and under the estimated best Normal distribution.  
dat.est<-data.frame(k=k,bin.prob=dbinom(k,sz,prop),  
 norm.prob=pnorm((k)+0.5,mu,sigma)-pnorm((k)-0.5,mu,sigma))  
# Make a column plot for the binomial probabilities showing the Normal approximation.  
  
ggplot(dat.est, aes(x=k))+geom\_col(aes(y=bin.prob))+  
 geom\_point(aes(y=norm.prob))+  
 stat\_function(fun=dnorm,args = list(mean =mu, sd = sigma))+  
 labs(title="Binomial(148,0.4994)")



The sum of squared differences in probabilities of the outcomes under the binomial model and under the Normal approximation:

q1.approx(sz,prop)

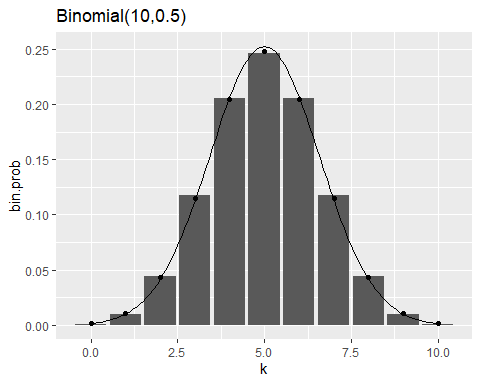
## [1] 3.048936e-08

## Q3, part 1

(5 points)

Please provide the corresponding visualization . What is the square error in approximating by a Normal distribution in this way?

N <- 10  
dist<- dbinom(x=1:N, size = N, prob = 0.5)  
normal.params<-normal.parameters.get(N,0.5)  
mu<-normal.params[1]  
sigma<-normal.params[2]  
  
# Plot only values of k that are at all likely.  
k<-qbinom(.00005,N,0.5):qbinom(.99995,N,0.5)  
  
# Make a data frame of these values of k, their probability under the binomial distribution, and under the estimated best Normal distribution.  
dist.est<-data.frame(k=k,bin.prob=dbinom(k,N,0.5),  
 norm.prob=pnorm((k)+0.5,mu,sigma)-pnorm((k)-0.5,mu,sigma))  
# Make a column plot for the binomial probabilities showing the Normal approximation.  
  
ggplot(dist.est, aes(x=k))+geom\_col(aes(y=bin.prob))+  
 geom\_point(aes(y=norm.prob))+  
 stat\_function(fun=dnorm,args = list(mean =mu, sd = sigma))+  
 labs(title="Binomial(10,0.5)")



# Square error:   
q1.approx(N,0.5)

## [1] 2.626771e-05

## Q3, part 2

(5 points)

What is the square error in approximating Binomial(10,0.1) by a Normal distribution in this way? Binomial(10,0.9)?

# In this case, we do not need the visualization, so we will only show the square error  
# Binomial(10,0.1)  
N <- 10  
prob <- 0.1  
## Square error:   
q1.approx(N,prob)

## [1] 0.01397547

# Binomial(10,0.9)  
N <- 10  
prob <- 0.9  
dist<- dbinom(x=1:N, size = N, prob = prob)  
normal.params<-normal.parameters.get(N,prob)  
## Square error:   
q1.approx(N,prob)

## [1] 0.01397547

# The expected outcome for this is for them to be equal, as 1 = 0.1+0.9

## Q3, part 3

(5 points)

What is the square error in approximating Binomial(100,0.5) by a Normal distribution in this way? Binomial(100,0.1)? Binomial(100,0.9)?

# In this case, we do not need the visualization, so we will only show the square error  
# Binomial(100,0.5)  
N <- 100  
prob <- 0.5  
## Square error:   
q1.approx(N,prob)

## [1] 8.108129e-08

# Binomial(100,0.1)  
N <- 100  
prob <- 0.1  
dist<- dbinom(x=1:N, size = N, prob = prob)  
normal.params<-normal.parameters.get(N,prob)  
## Square error:   
q1.approx(N,prob)

## [1] 0.0003586778

# Binomial(100,0.9)  
N <- 100  
prob <- 0.9  
dist<- dbinom(x=1:N, size = N, prob = prob)  
normal.params<-normal.parameters.get(N,prob)  
## Square error:   
q1.approx(N,prob)

## [1] 0.0003586778

## Q3, part 4

(5 points)

For a fixed value of , what condition on the of distribution makes the Normal approximation a better approximation? For a fixed value of , what conditions on the of distribution makes the Normal approximation a better approximation? Please feel free to examine the results of values for the size and probability beyond those considered above.

What can you say about the size of the error when the value of is large and the value of is near 0.5?

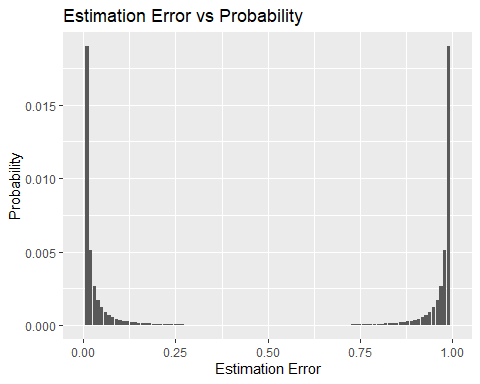
# Fixed value of size, variable probability to reduce the error  
N<-100  
Prob <- seq(from=0, to = 1, by= 0.01)  
vec<-c() # We create an empty vector  
# Then, as q1.approx does not accept a list of elements as a parameter, we have to use a for loop  
for ( val in Prob){  
 vec<-c(vec, q1.approx(N, val))  
}  
#We sort it   
vecSorted<-sort(vec)  
# Then we can look for the values with the highest error and their position in the vector, which can be directly related to the probability used to get this error  
highest<-match(tail(vecSorted, 10),vec)  
# Probabilities used:  
highest\*0.01

## [1] 0.06 0.96 0.05 0.97 0.04 0.98 0.03 0.99 0.02 1.00

# lowest error  
lowest<-match(head(vecSorted,10),vec)  
lowest\*0.01

## [1] 0.01 0.01 0.51 0.50 0.52 0.53 0.49 0.48 0.54 0.55

# With this, we can see that the highest error happens in the edges of the distribution, and the lowest error happens in the middle of the distribution  
ggplot(data.frame(vec),aes(x=Prob,y=vec))+  
 geom\_col()+  
 labs(title="Estimation Error vs Probability")+  
 xlab("Estimation Error")+  
 ylab("Probability")



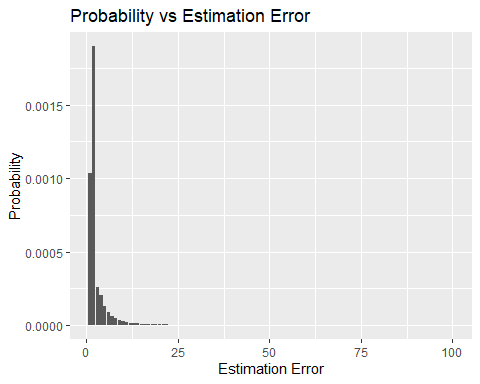
# Fixed probability, variable size  
Prob<-0.5  
N <- seq(from=1, to = 100, by= 1)  
vec<-c() # We create an empty vector  
# Then, as q1.approx does not accept a list of elements as a parameter, we have to use a for loop  
for ( val in N){  
 vec<-c(vec, q1.approx(val, Prob))  
}  
#We sort it   
vecSorted<-sort(vec)  
# Then we can look for the values with the highest error and their position in the vector, which can be directly related to the N used to get this error  
highest<-match(tail(vecSorted, 10),vec)  
# N:  
highest

## [1] 10 9 8 7 6 5 4 3 1 2

# lowest error  
lowest<-match(head(vecSorted,10),vec)  
lowest

## [1] 100 99 98 97 96 95 94 93 92 91

# With this, we can see that the highest error happens within the first few elements, meaning that lower numbers = higher error.  
ggplot(data.frame(vec),aes(x=N,y=vec))+  
 geom\_col()+  
 labs(title="Probability vs Estimation Error")+  
 xlab("Estimation Error")+  
 ylab("Probability")



For the final question, the previous graph shows us that with a high value of with equal to 0.5, the error is very low. We can see that with a value of 100 and a probability of 0.5, the error is:

q1.approx(100,0.5)

## [1] 8.108129e-08

while with n = 1000:

q1.approx(1000,0.5)

## [1] 2.556252e-10

The higher the n, the lower the error