5320\_final\_project

Milagros Crisp

2022-05-12

# with two candidates with one 20 percentage points over implies lower and end points to be .40 and .60 suggesting mean = .5  
# where .50 + 2\*STD = .6  
# hence the standard deviation is .05  
  
# found in substituting that alpha/(alpha + beta) = .5   
# which implied that alpha = beta   
  
# and by using variance alpha\*beta / (alpha + beta)^2 \* (alpha + beta + 1) = .0025  
# substitituting beta for alpha we get that   
# beta = 49.5 = alpha  
  
qbeta(c(.025, .975), 49.5,49.5)

## [1] 0.4022148 0.5977852

# question 1

N <- 800  
alpha <- 49.5  
beta <- 49.5  
  
y <- 52  
n <- 95  
  
alpha\_post <- alpha + y  
beta\_post <- beta + n - y  
  
# the size of our posterior sample:  
S <- 10000  
  
# take a random sample from the posterior distribution:  
pi\_s <- rbeta(S,alpha\_post,beta\_post)  
  
# let's estimate E[pi|y], which is the posterior mean of pi, simply by looking at the sample mean of our  
# posterior sample:  
mean(pi\_s)

## [1] 0.5234336

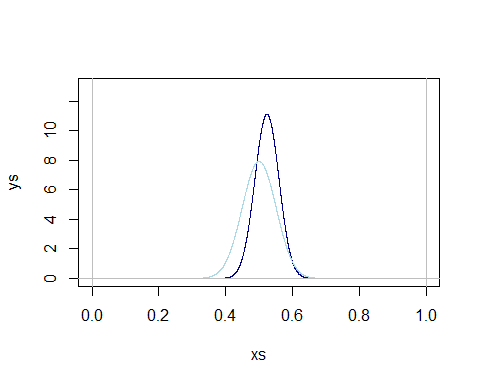
# compare to the exact value:  
alpha\_post / (alpha\_post + beta\_post)

## [1] 0.5231959

# what is the posterior probability that pi is bigger than 50%?  
mean(pi\_s > 0.5)

## [1] 0.7449

xs <- seq(0,12,.1)  
draw\_beta(alpha\_post, beta\_post, clr = "darkblue")  
draw\_beta(alpha, beta, new = FALSE, clr = "lightblue")



*part 1b*

eta\_min <- log(.4/(1-.4))  
eta\_max <- log(.6/(1-.6))  
  
  
# this implies that mu is zero   
c(eta\_min, eta\_max)

## [1] -0.4054651 0.4054651

# so then standard deviation from mu = 0 is   
std <- eta\_max/2  
std

## [1] 0.2027326

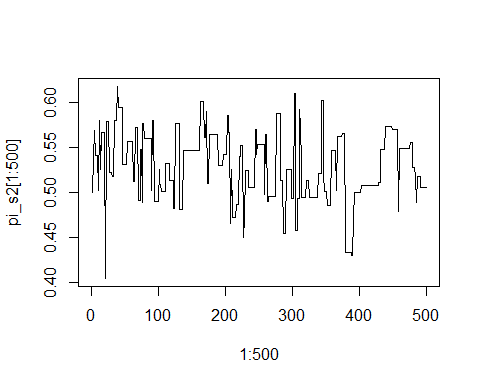
# ----- METROPOLIS ALGORITHM FOR SAMPLING FROM POSTERIOR FOR NORMAL MODEL -----  
  
  
# number of chain iterations:  
S <- 100000  
  
# mu values for all S elements in the chain:  
eta\_s <- numeric(S)  
  
  
# parameters for the prior with an adjusted standard deviation  
std\_dev <- 0.3   
mu\_0 <- 0  
y <- 52  
n <- 95  
  
# initializing an empty eta  
eta <- 0  
  
# a vector chain for all eta\_s  
eta\_s <- numeric(S)  
  
  
  
for (s in 1:S) {  
  
 # propose a new value for the chain:  
 eta\_star <- eta + rnorm(1)  
  
 # prior and likelihood ratios  
 prior\_density\_ratio <- dnorm(eta\_star,mu\_0 ,2.4\*std\_dev^2) /   
 dnorm(eta, mu\_0, 2.4\*std\_dev^2)  
   
 likelihood\_ratio <- (dbinom(y, n,(exp(eta\_star))/(1+exp(eta\_star)))) / (dbinom(y,  
 n,(exp(eta))/(1+exp(eta))))  
  
 # accept with the appropriate probability:  
 if (runif(1) < prior\_density\_ratio\*likelihood\_ratio) {  
 eta <- eta\_star  
 }  
  
 # save the next x in the chain:  
 eta\_s[s] <- eta  
  
}  
  
pi\_s2 <- exp(eta\_s)/(1+exp(eta\_s))  
  
# effective sample size and mcm efficiency  
  
effectiveSize(pi\_s2)

## var1   
## 11951.77

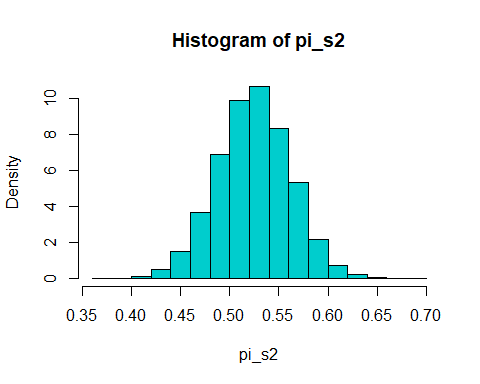
effectiveSize(pi\_s2)/S

## var1   
## 0.1195177

# generate a trace plot of the sample:  
  
plot(1:500,pi\_s2[1:500],type="l")



# histogram of posterior samples:  
xs <- seq(0,12,.1)  
hist(pi\_s2,prob=TRUE, col = "cyan3")

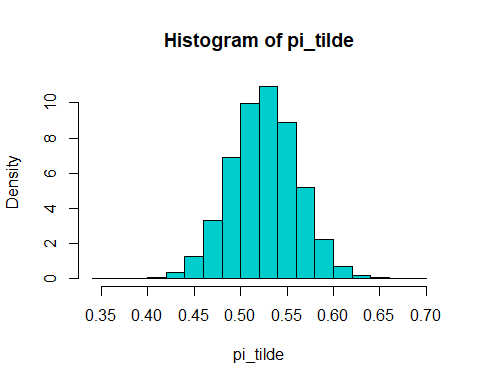


mean(pi\_s2 > .5)

## [1] 0.74633

# question 2

# generate random pis  
r\_pis <- rbeta(S, alpha\_post, beta\_post)  
r\_y <- rbinom(S,N - n, r\_pis)  
  
pi\_tilde <- (y + r\_y)/N  
hist(pi\_tilde, prob = TRUE, col = "cyan3")



mean(pi\_tilde > .5)

## [1] 0.76305

mean(pi\_tilde == .5)

## [1] 0.01082

# question 3

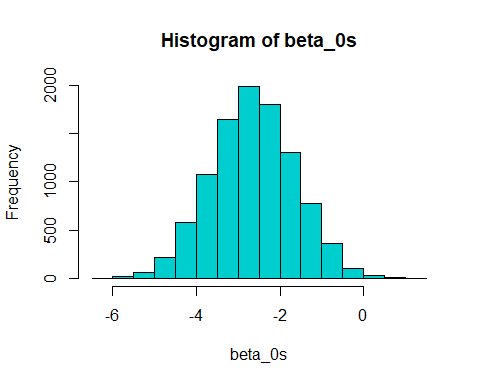
# Use the Metropolis algorithm to sample from the joint posterior distribution of β0 and β1. Confirm your results are accurate by comparing them to frequentist estimates of this logistic regression model (hint: glm(vote ~ age,binomial)  
  
age <- voter\_roll$age[!is.na(voter\_roll$vote)]  
vote <- voter\_roll$vote[!is.na(voter\_roll$vote)]  
  
# referencing what we're supposed to estimate  
  
glm <- glm(vote ~ age, data = voter\_roll, family = binomial)  
coef <- coef(glm)  
beta\_0 <- coef[1] # gonna start with these values   
beta\_1 <- coef[2]  
  
# number of chain iterations  
S <- 10000  
  
# initializing mu\_s values for all S elements in the chain:  
beta\_0s <- numeric(S)  
beta\_1s <- numeric(S)  
  
for (s in 1:S) {  
  
 # propose a new value for the chain:  
 beta\_0\_star <- beta\_0 + rnorm(1)  
 beta\_1\_star <- beta\_1 + rnorm(1)  
   
 eta <- beta\_0 + beta\_1\*age  
 eta\_star <- beta\_0\_star + beta\_1\_star\*age  
   
 pi <- exp(eta)/(1+exp(eta))  
 pi\_star <- exp(eta\_star)/(1+exp(eta\_star))  
   
 likelihood\_ratio <- prod(dbinom(vote,1,pi\_star))/prod(dbinom(vote,1,pi))  
   
 # accept with the appropriate probability:  
 if (runif(1) < likelihood\_ratio) {  
 beta\_not\_est <- beta\_0\_star  
 beta\_one\_est <- beta\_1\_star  
 }  
   
 # save the next x in the chain:  
 beta\_0s[s] <- beta\_0\_star  
 beta\_1s[s] <- beta\_1\_star  
}  
  
mean(beta\_0s)

## [1] -2.64821

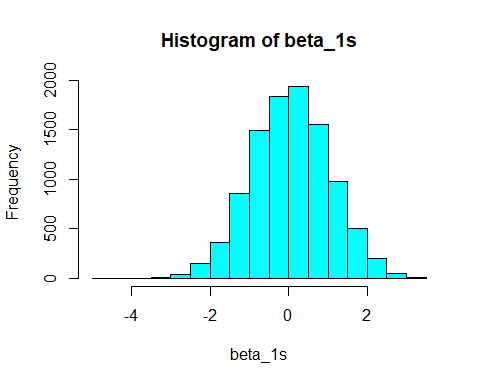
mean(beta\_1s)

## [1] 0.06005053

hist(beta\_0s, col = "cyan3")



hist(beta\_1s, col = "cyan")



#let ̃π be the proportion of the N = 800 voters in Backwoodsville who support Gray. We have ̃π = (y0 +∑yi)/N,  
  
# where y0 is the count of the n voters in the sample who suppor Gray, and the sum is over the N−n = 800−95 = 705 voters not included in the sample,   
  
y0 <- sum(vote)  
N\_n <- 705  
  
# and yi ∼ Bin(1,πi). Find the posterior predictive distribution for ̃π, and compare this to your results in Part 2.  
  
yi <- rbinom(N\_n, 1, pi)  
  
pi\_tilde3 <- (y0 + sum(yi))/ N\_n  
pi\_tilde3

## [1] 0.6340426

This turned out to be a lower estimate compared to the 0.76322 in question 2.

# question 4

In conclusion, it appears that we can estimate that the proportion of poeple voting for Gary Gray is 74.15%. Choosing 800 observations to represent the town of backwoodsville we estimate that the proportion of the townsfolk to vote for Gary Gray is 76.23%. Using a metropolis algorithm we managed to estimate that for every 10 years that a person agaes in the town Backswoodville the more likely they are to vote through the phone.