5320_final_project

Milagros Crisp

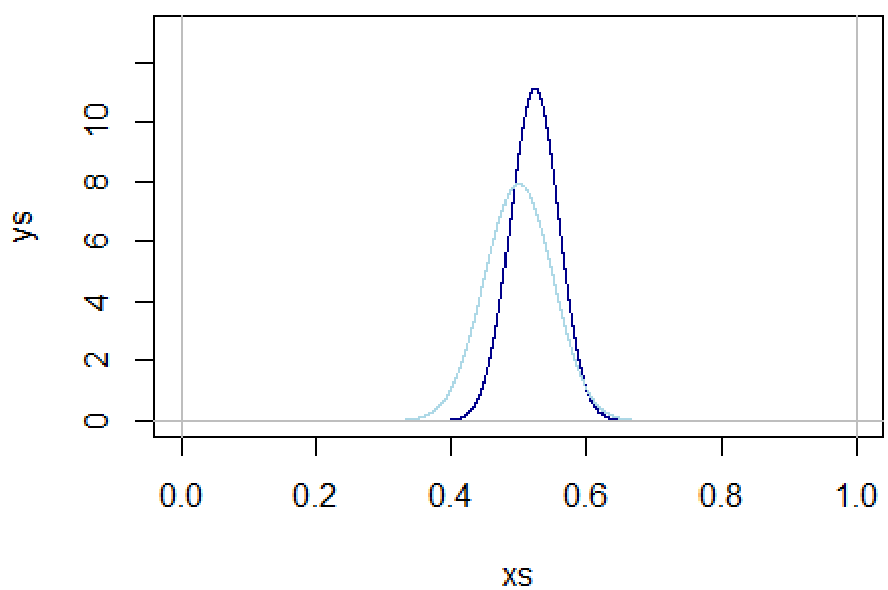
2022-05-12

```
# 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
```

```r
# 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")
```

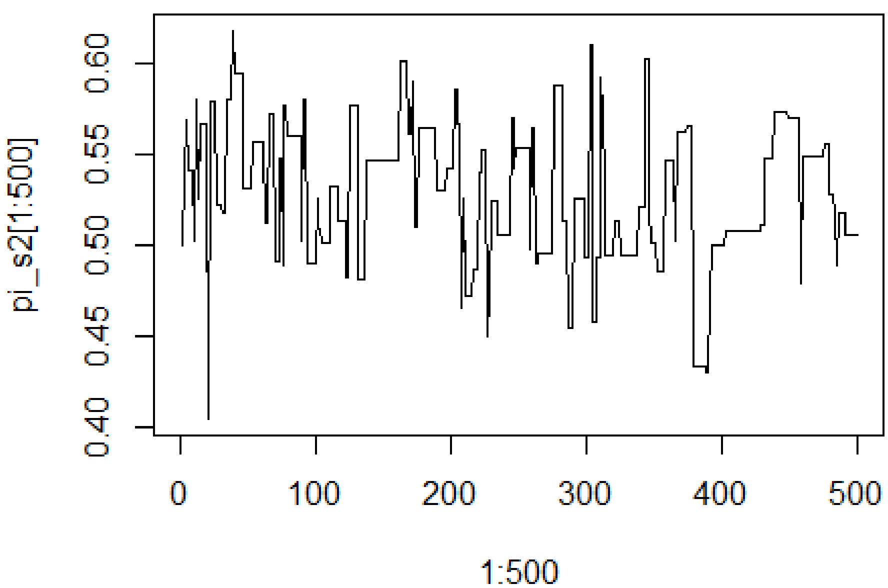


part 1b

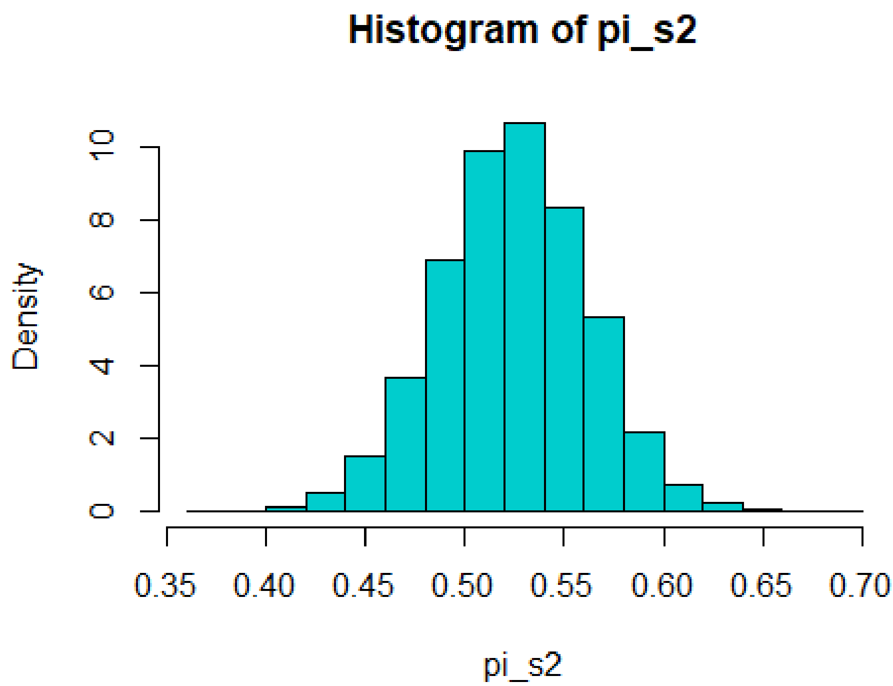
```r
eta_min <- log(.4/(1-.4))
eta_max <- log(.6/(1-.6))


# 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")
```



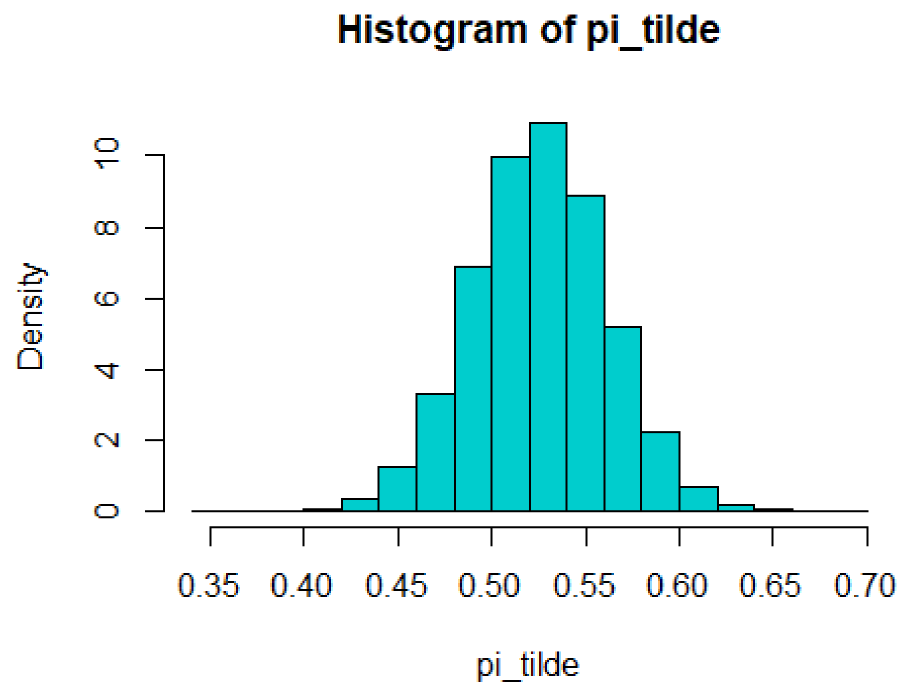

```
# histogram of posterior samples:
xs <- seq(0,12,.1)
hist(pi_s2,prob=TRUE, col = "cyan3")
```




```
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")
```

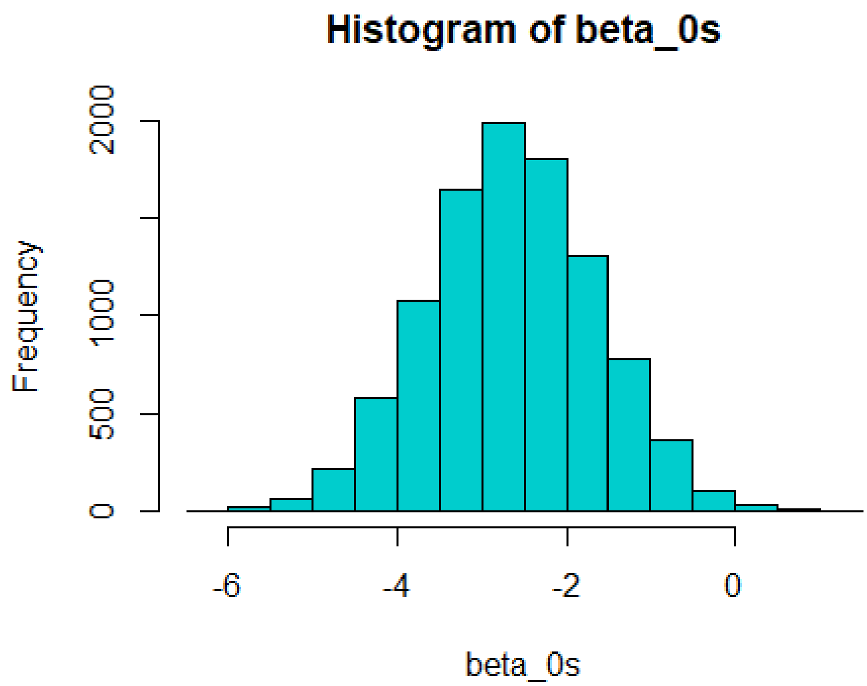



```
mean(pi_tilde > .5)

## [1] 0.76305

mean(pi_tilde == .5)

## [1] 0.01082
```

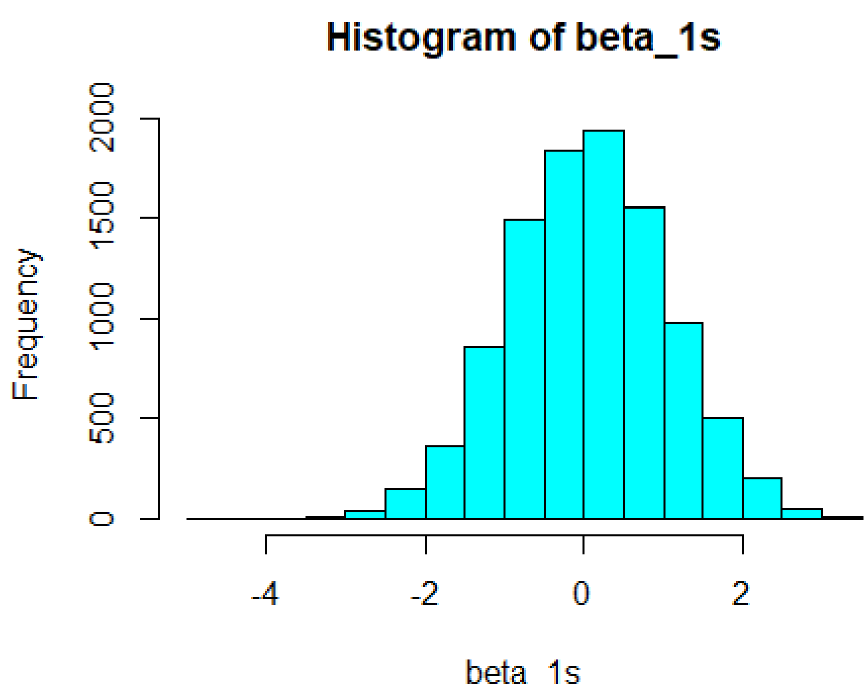
question 3

```
# 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
```

```r
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")
```



```r
#let  π̃ be the proportion of the N = 800 voters in Backwoodsville who support Gray. We have  π̃ = (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  π̃, 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 people 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.