Bayesian Statistics- HW3

魏上傑

2023-04-23

1.

(a)

Alex believes there will be 8 successes and 2 failures, corresponding to the Beta(8,2).

(b)

```
# Benedit believes the 0.2 quantile of the prior is 0.3
# and the 0.9 quantile of the prior is 0.4
beta.select(list(p=0.2, x=0.3), list(p=0.9, x=0.4))
```

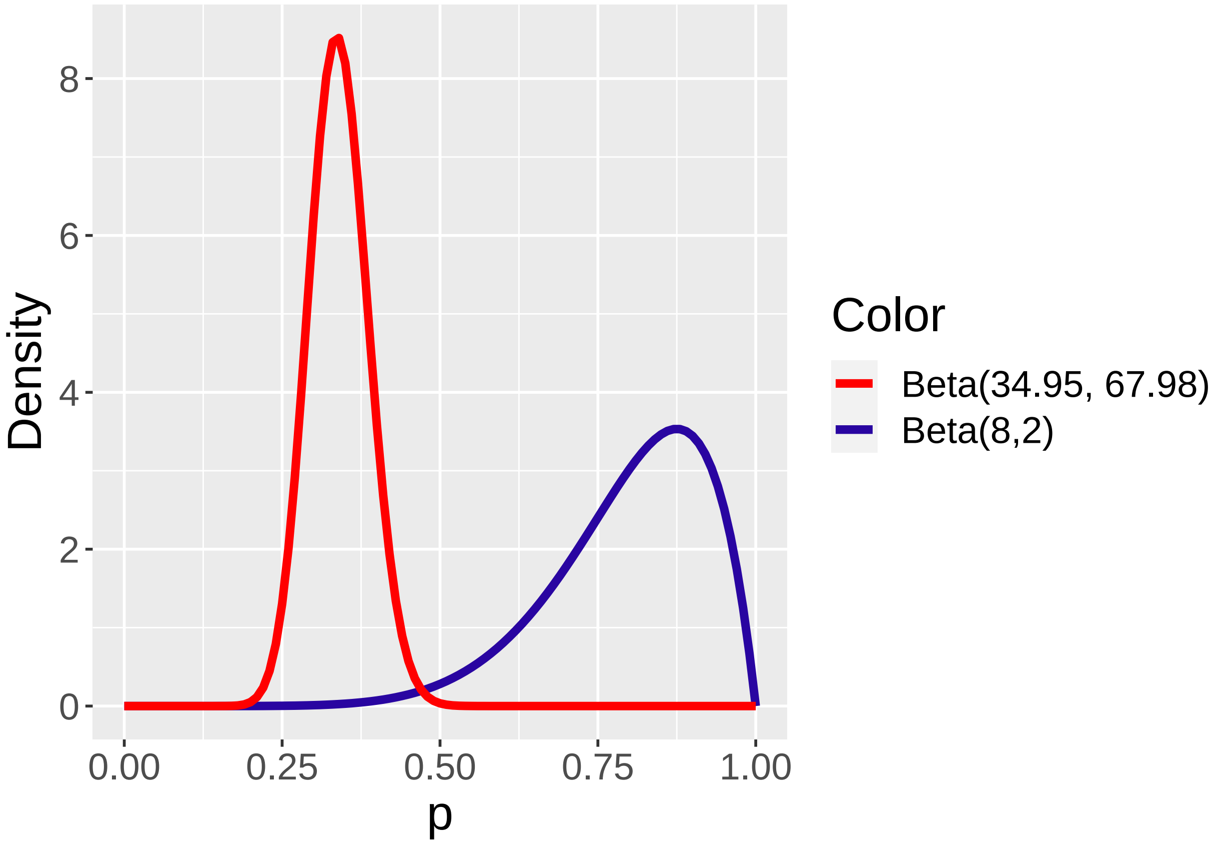
```
## [1] 34.95 67.98
```

Thus, we will use Beta(34.95, 67.98) for Benedit's prior.

(c)

```
ggplot(data.frame(x=c(0,1)), aes(x)) +
  stat_function(fun = dbeta, geom="line",
                aes(color = "Beta(8,2)"),
                linetype= "solid",
                linewidth=1.5,
                args = list(shape1=8,
                            shape2=2)) +
  stat_function(fun = dbeta, geom="line",
                aes(color="Beta(34.95, 67.98)"),
                linetype= "solid",
                linewidth=1.5,
                args = list(shape1=34.95,
```

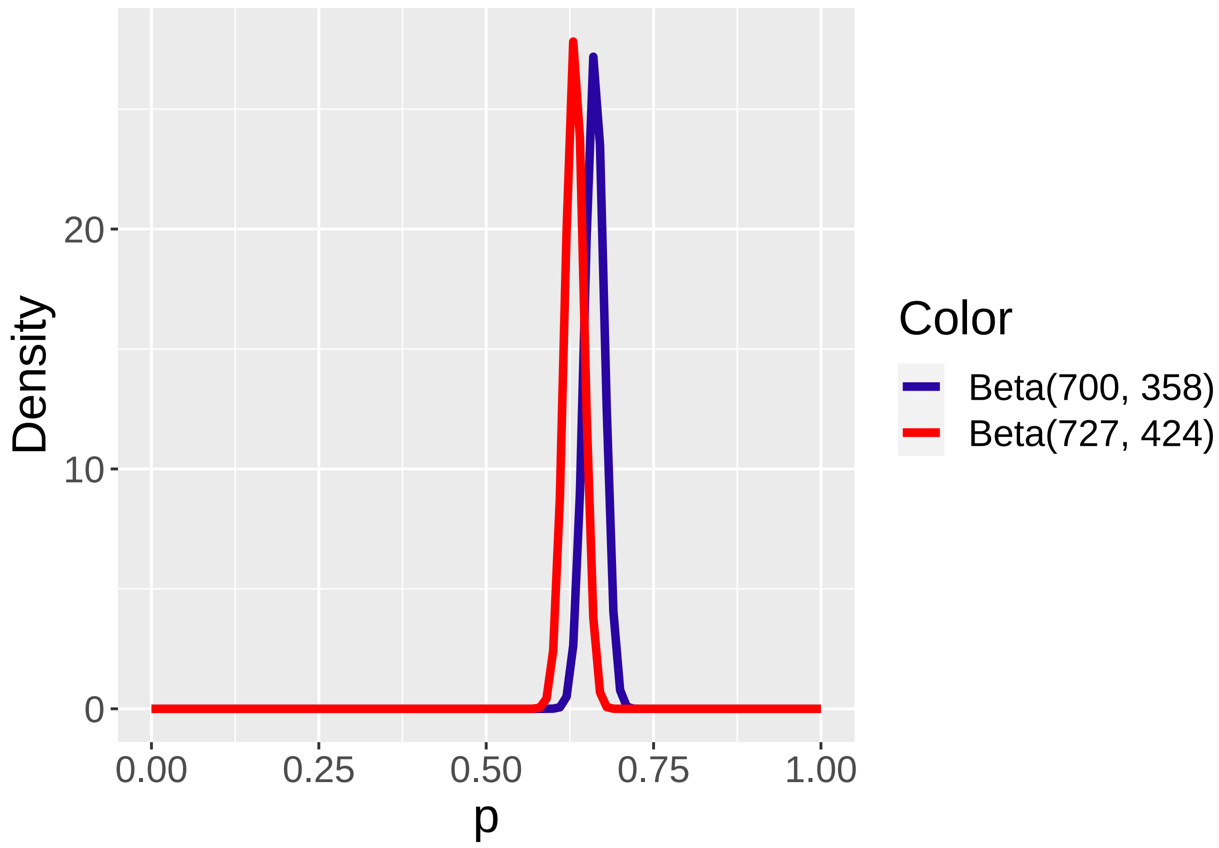
```
                          shape2=67.98)) +
  xlab("p") + ylab("Density") +
  scale_color_manual(name="Color",
                     values=c("Beta(8,2)"=crcblue,
                              "Beta(34.95, 67.98)"= "red")) +
  increasefont()
```



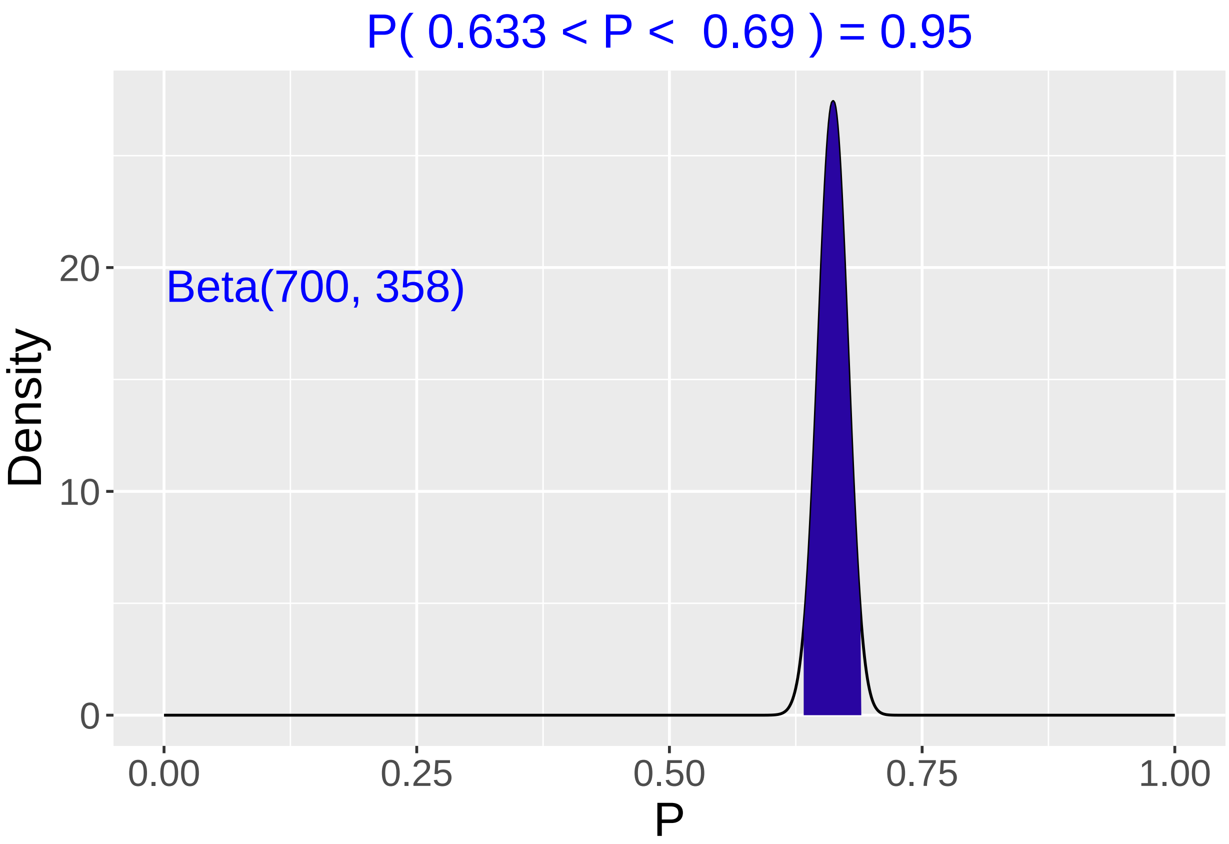

```
# Calculate the posterior
Alex_ab <- c(8,2)
Ben_ab <- c(34.95, 67.98)
yny <- c(692, 1048-692)
Alex_ab_new <- Alex_ab + yny
Ben_ab_new <- Ben_ab + yny
```

```
ggplot(data.frame(x=c(0,1)), aes(x)) +
  stat_function(fun = dbeta, geom="line",
                aes(color = "Beta(700, 358)"),
                linetype= "solid",
                linewidth=1.5,
                args = list(shape1=700,
                            shape2=358)) +
  stat_function(fun = dbeta, geom="line",
```

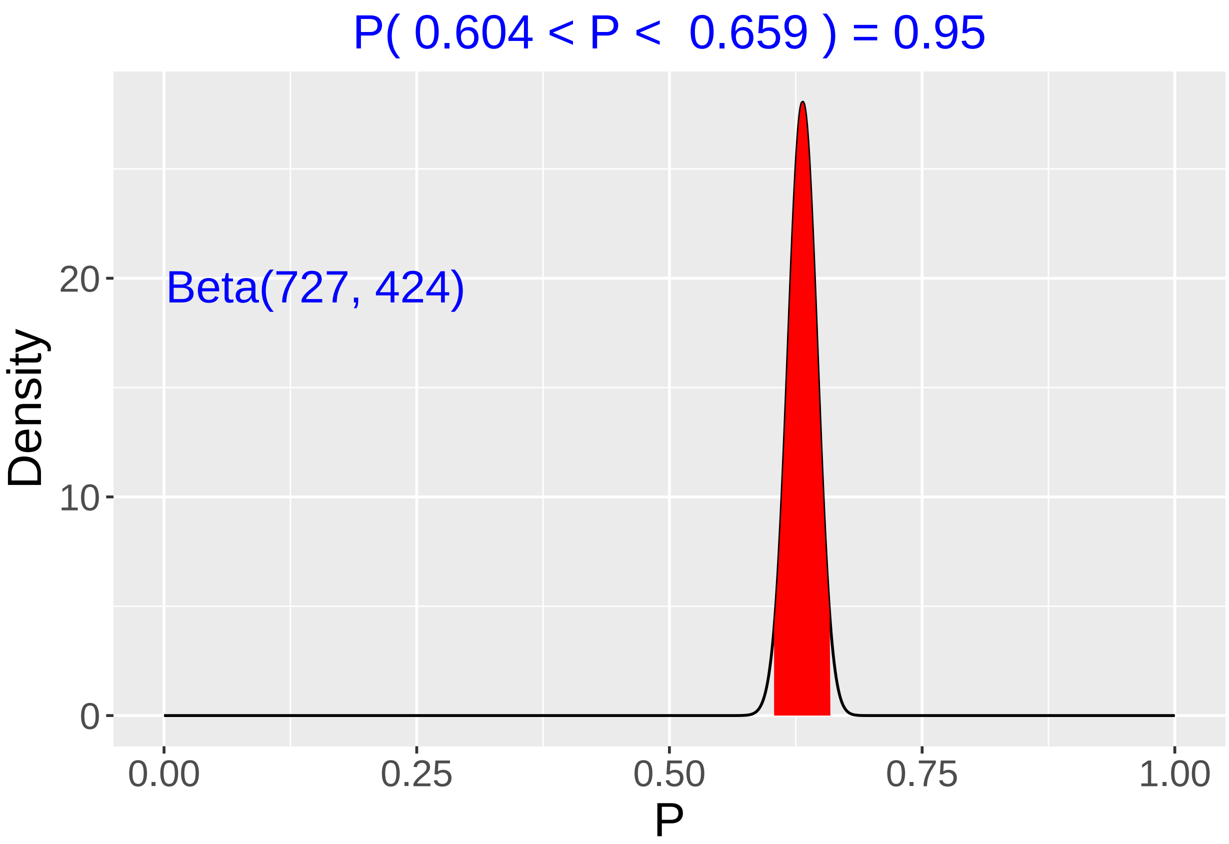
```
                aes(color="Beta(727, 424)"),
                linetype= "solid",
                linewidth=1.5,
                args = list(shape1=727,
                            shape2=424)) +
  xlab("p") + ylab("Density") +
  scale_color_manual(name="Color",
                     values=c("Beta(700, 358)"=crcblue,
                              "Beta(727, 424)"= "red")) +
  increasefont()
```




```
beta_interval(0.95, c(700, 358), Color= crcblue) +
  theme(text = element_text(size=18))
```




```r
beta_interval(0.95, c(727, 424), Color= "red") +
  theme(text = element_text(size=18))
```



(d)

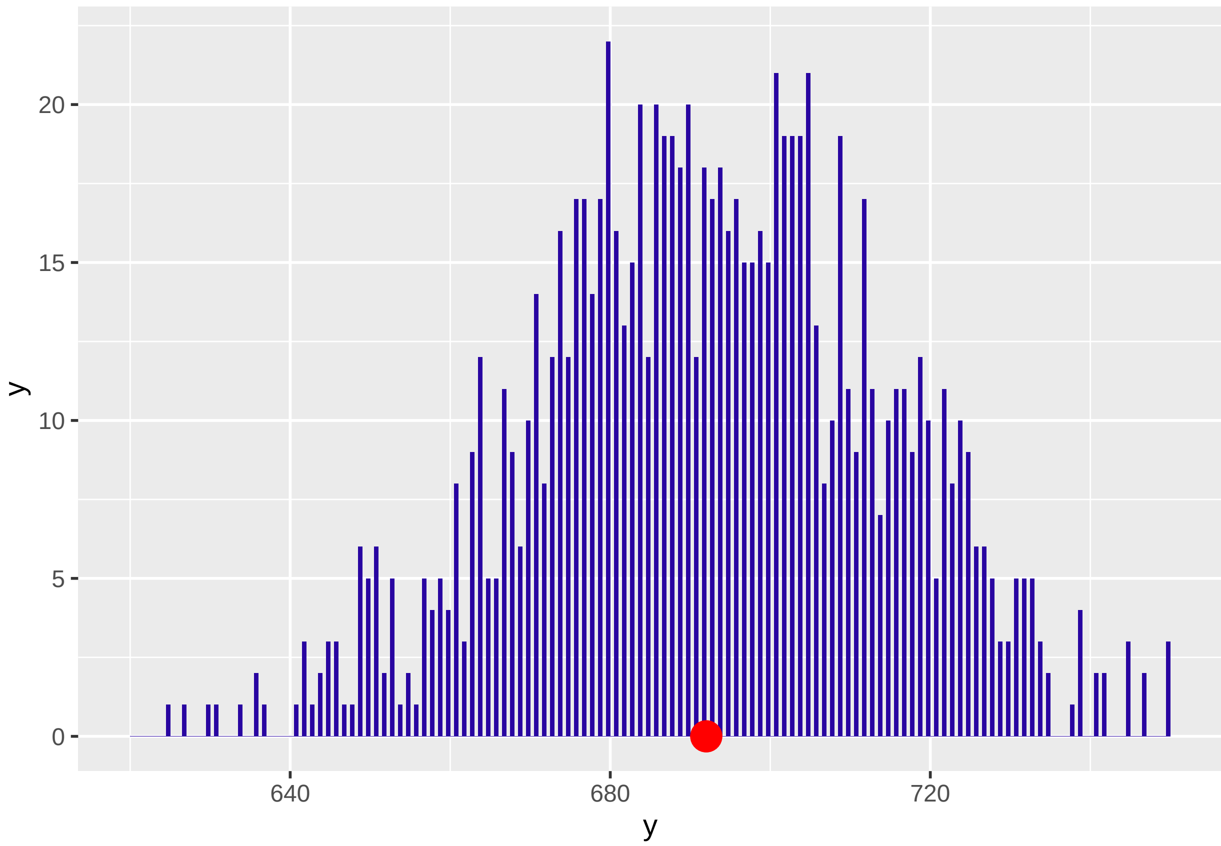
```r
# prior preditive checks for Alex
S <- 1000
a <- 8; b <- 2
n <- 1048; y <- 692

newy <- as.data.frame(rep(NA, S))
names(newy)= c("y")

for (s in 1:S){
  pred_p_sim <- rbeta(1, a+y, b+n-y)
  pred_y_sim <- rbinom(1, n, pred_p_sim)
  newy[s,] = pred_y_sim
}

ggplot(data=newy, aes(newy$y))+
  geom_histogram(breaks=seq(620, 750, by=0.5), fill = crcblue)+
  annotate("point", x=692, y=0, colour = "red", size =5)+
  xlab("y")
```

```
## Warning: Use of `newy$y` is discouraged.
## i Use `y` instead.
```



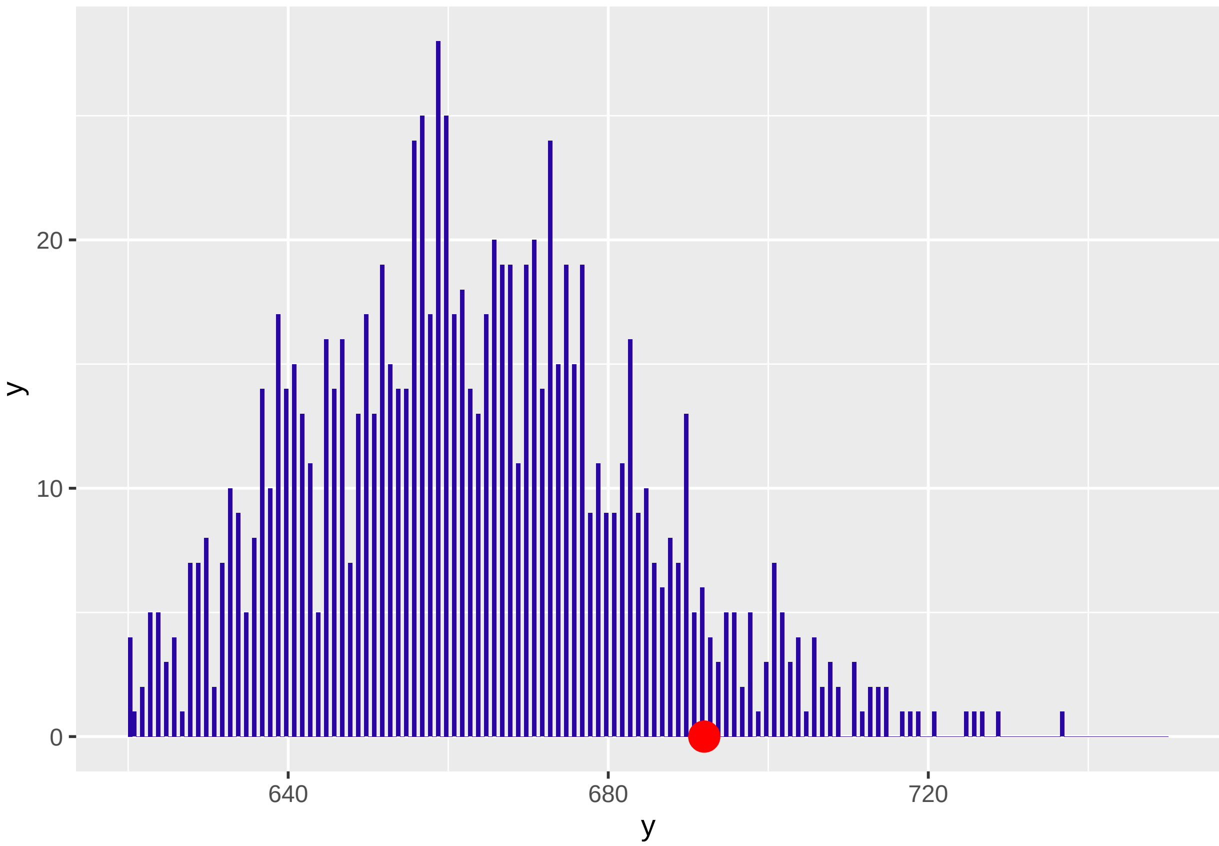
```
# prior preditive checks for Ben
S <- 1000
a <- 34.95; b <- 67.98
n <- 1048; y <- 692

newy <- as.data.frame(rep(NA, S))
names(newy)= c("y")

for (s in 1:S){
  pred_p_sim <- rbeta(1, a+y, b+n-y)
  pred_y_sim <- rbinom(1, n, pred_p_sim)
  newy[s,] = pred_y_sim
}

ggplot(data=newy, aes(newy$y))+
  geom_histogram(breaks=seq(620, 750, by=0.5), fill = crcblue)+
  annotate("point", x=692, y=0, colour = "red", size =5)+
  xlab("y")
```

```
## Warning: Use of `newy$y` is discouraged.
## i Use `y` instead.
```



From the prior predictive checks, we conclude that Alex's prior is more appropriate for this teenagers and television data since the red dot is closer to Alex's model.

2.

(a)

```
S <- 1000
a <- 8; b <- 2
n <- 1048; y <- 692
BetaSamples <- rbeta(S, a+y, b+n-y)
odds <- BetaSamples/ (1-BetaSamples)
mean(odds)
```

```
## [1] 1.957605
```

```
median(odds)
```

```
## [1] 1.953653
```

```
quantile(odds, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 1.725181 2.211119
```

(b)

```
S <- 1000
a <- 34.95; b <- 67.98
n <- 1048; y <- 692
BetaSamples <- rbeta(S, a+y, b+n-y)
odds <- BetaSamples/ (1-BetaSamples)
mean(odds)
```

```
## [1] 1.71919
```

```
median(odds)
```

```
## [1] 1.719706
```

```
quantile(odds, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 1.523884 1.945950
```

(c) As above.

3.

```
set.seed(123)  # for reproducibility
alpha <- 15.06
beta <- 10.56
S <- c(10, 100, 500, 1000, 5000)
results <- list()

for (s in S) {
  ps <- rbeta(s, alpha, beta)  # generate s random samples from Beta distribution
  lower <- quantile(ps, 0.05)  # find 5th percentile
  upper <- quantile(ps, 0.95)  # find 95th percentile
  results[[as.character(s)]] <- c(lower, upper)  # store credible interval
}

# print results
for (s in S) {
  cat("Middle 90% credible interval for S =", s, "is [", round(results[[as.character(s)]][
}
```

```
## Middle 90% credible interval for S = 10 is [ 0.382 , 0.656 ]
## Middle 90% credible interval for S = 100 is [ 0.45 , 0.721 ]
## Middle 90% credible interval for S = 500 is [ 0.427 , 0.732 ]
## Middle 90% credible interval for S = 1000 is [ 0.42 , 0.742 ]
## Middle 90% credible interval for S = 5000 is [ 0.429 , 0.741 ]
```

As we can see from the results, the simulated credible intervals become narrower as the simulation size S increases. This is because larger sample sizes result in more precise estimates of the true distribution, and thus the credible intervals become more accurate. However, we can also see that even for small sample sizes like S = 10, the simulated credible interval still contains the true posterior interval estimate of [0.427, 0.741], although it is wider than the intervals obtained with larger sample sizes.

In summary, simulating a large number of random samples results in more precise credible intervals, but even with small sample sizes, it is still possible to obtain credible intervals that contain the true posterior interval estimate.

4.

(a)

$$L(\mu_A) = \prod_{i=1}^{n_A} \frac{1}{4\sqrt{2\pi}} exp(-\frac{(y_{A,i} - \mu_A)^2}{2 \times 4^2}) \tag{1}$$
$$= (4^2 2\pi)^{-\frac{n_A}{2}} exp(-\frac{\sum(y_{A,i} - \mu_A)^2}{2 \times 4^2}) \tag{2}$$

$$L(\mu_N) = \prod_{j=1}^{n_N} \frac{1}{4\sqrt{2\pi}} exp(-\frac{(y_{N,j} - \mu_N)^2}{2 \times 4^2}) \tag{3}$$
$$= (4^2 2\pi)^{-\frac{n_N}{2}} exp(-\frac{\sum(y_{N,j} - \mu_N)^2}{2 \times 4^2}) \tag{4}$$

(b)

Since the prior belief about μ_A and μ_N are independent, the posterior distributions for μ_A and μ_N are independent as well.

- The prior distribution

$$\pi(\mu_A) = \frac{1}{\sqrt{2\pi}\sigma_A} exp(-\frac{(\mu_A - \gamma_A)^2}{2\sigma_A^2}) \tag{5}$$
$$= \frac{1}{\sqrt{2\pi}} \phi_A^{\frac{1}{2}} exp(-\frac{\phi_A}{2}(\mu_A - \gamma_A)^2) \tag{6}$$

$$\pi(\mu_N) = \frac{1}{\sqrt{2\pi}\sigma_N} exp(-\frac{(\mu_N - \gamma_N)^2}{2\sigma_N^2}) \tag{7}$$
$$= \frac{1}{\sqrt{2\pi}} \phi_N^{\frac{1}{2}} exp(-\frac{\phi_N}{2}(\mu_N - \gamma_N)^2) \tag{8}$$

- The posterior distribution

$$\pi(\mu_A | y_1, ..., y_{n_A}, \sigma = \frac{1}{\sqrt{\phi}} = 4) \propto \pi(\mu_A) L(\mu_A) \tag{9}$$
$$\propto exp(-\frac{\phi_A}{2}(\mu_A - \gamma_A)^2) exp(-\frac{\sum(y_{A,i} - \mu_A)^2}{2 \times 4^2}) \tag{10}$$
$$\propto exp(-\frac{1}{2}(\phi_A + n_A\phi)\mu_A^2 + \frac{1}{2}(2\phi_A\gamma_A + 2n_A\phi\bar{y}_A)\mu_A) \tag{11}$$
$$\propto exp(\frac{1}{2}(\phi_A + n_A\phi)(\mu_A - \frac{\phi_A\gamma_A + n_A\phi\bar{y}_A}{\phi_A + n_A\phi})^2) \tag{12}$$

Hence, this is a Normal density with mean $\frac{\phi_A\gamma_A+n_A\phi\bar{y}_A}{\phi_A+n_A\phi}$ and std.dev $\frac{1}{\sqrt{\phi_A+n_A\phi}}$

$$\pi(\mu_N|y_1,...,y_{n_N},\sigma=\frac{1}{\sqrt{\phi}}=4)\propto\pi(\mu_N)L(\mu_N) \tag{13}$$

$$\propto exp(-\frac{\phi_N}{2}(\mu_N-\gamma_N)^2)exp(-\frac{\sum(y_{N,i}-\mu_N)^2}{2\times 4^2}) \tag{14}$$

$$\propto exp(-\frac{1}{2}(\phi_N+n_N\phi)\mu_N^2+\frac{1}{2}(2\phi_N\gamma_N+2n_N\phi\bar{y}_N)\mu_N) \tag{15}$$

$$\propto exp(\frac{1}{2}(\phi_N+n_N\phi)(\mu_N-\frac{\phi_N\gamma_N+n_N\phi\bar{y}_N}{\phi_N+n_N\phi})^2) \tag{16}$$

Hence, this is a Normal density with mean $\frac{\phi_N\gamma_N+n_N\phi\bar{y}_N}{\phi_N+n_N\phi}$ and std.dev $\frac{1}{\sqrt{\phi_N+n_N\phi}}$

5.

(a)

```
gamma_A <- 0
sigma_A <- 20
phi_A <- 1/sigma_A^2
ybar_A <- 15.2
phi <- 1/4^2 # sigma=4
n_A <- 6
mu_A <- (phi_A*gamma_A+n_A*ybar_A*phi)/(phi_A+n_A*phi)
sd_A <- sqrt(1/(phi_A+n_A*phi))

S <- 5000

s_A <- rnorm(S, mu_A, sd_A)
```

```
gamma_N <- 0
sigma_N <- 20
phi_N <- 1/sigma_N^2
ybar_N <- 6.2
phi <- 1/(4^2) # sigma=4
n_N <- 6
mu_N <- (phi_N*gamma_N+n_N*ybar_N*phi)/(phi_N+n_N*phi)
sd_N <- sqrt(1/(phi_N+n_N*phi))

S <- 5000

s_N <- rnorm(S, mu_N, sd_N)
```

```r
d <- s_A-s_N
sum(d>0)/S
```

```
## [1] 1
```

(b)

```r
sigma <- 4
pred_mu_A_sim <- rnorm(1, mu_A, sd_A)
pred_y_A_sim <- rnorm(n_A, pred_mu_A_sim, sigma)

pred_mu_N_sim <- rnorm(1, mu_N, sd_N)
pred_y_N_sim <- rnorm(n_N, pred_mu_N_sim, sigma)

S <- 6
d <- pred_y_A_sim-pred_y_N_sim
sum(d>0)/S
```

```
## [1] 0.8333333
```