STAT404_HW1_2017150431

CWY

2022-10-15

1.(a) #3.1

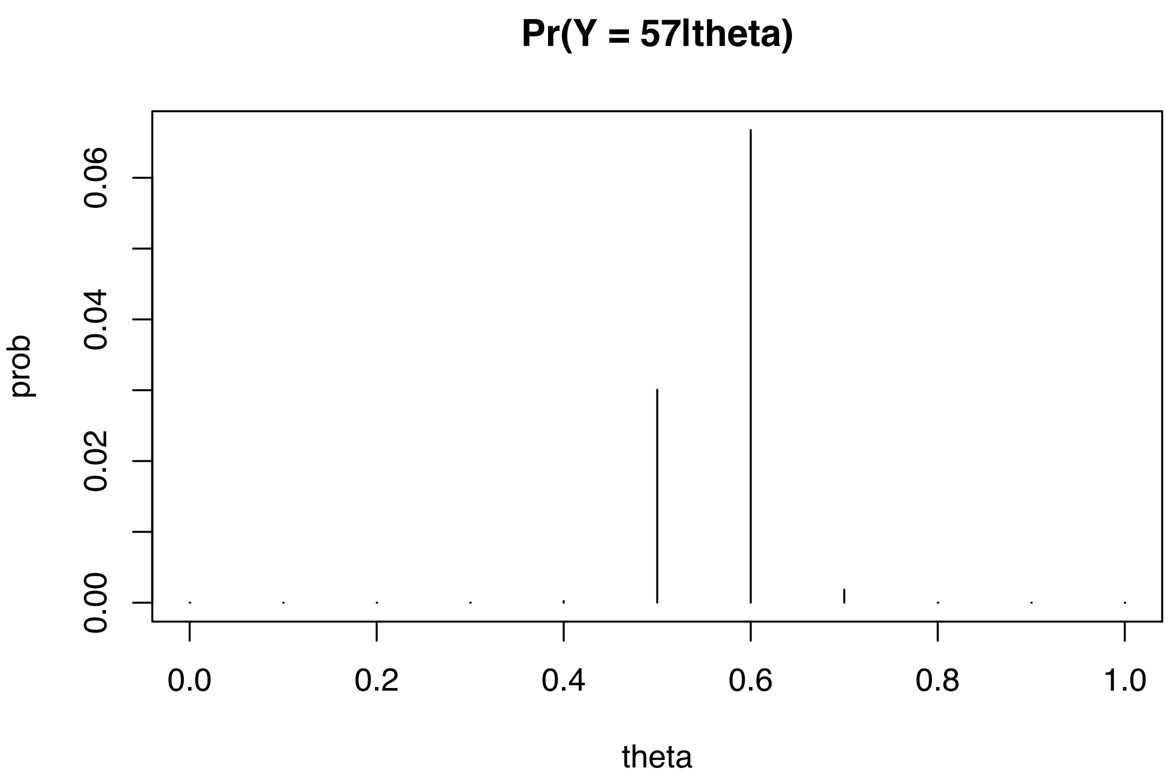
a) $Pr(Y_1=y_1,\dots,Y_{100}=y_{100}|\theta)=\prod_{i=1}^{100}\theta^{y_i}(1-\theta)^{1-y_i}1_{\{0,1\}}(y_i)$

$Pr(Y_i=y|\theta)=\binom{100}{y}\theta^{y}(1-\theta)^{100-y}$

1-(a) # 3.1

b)

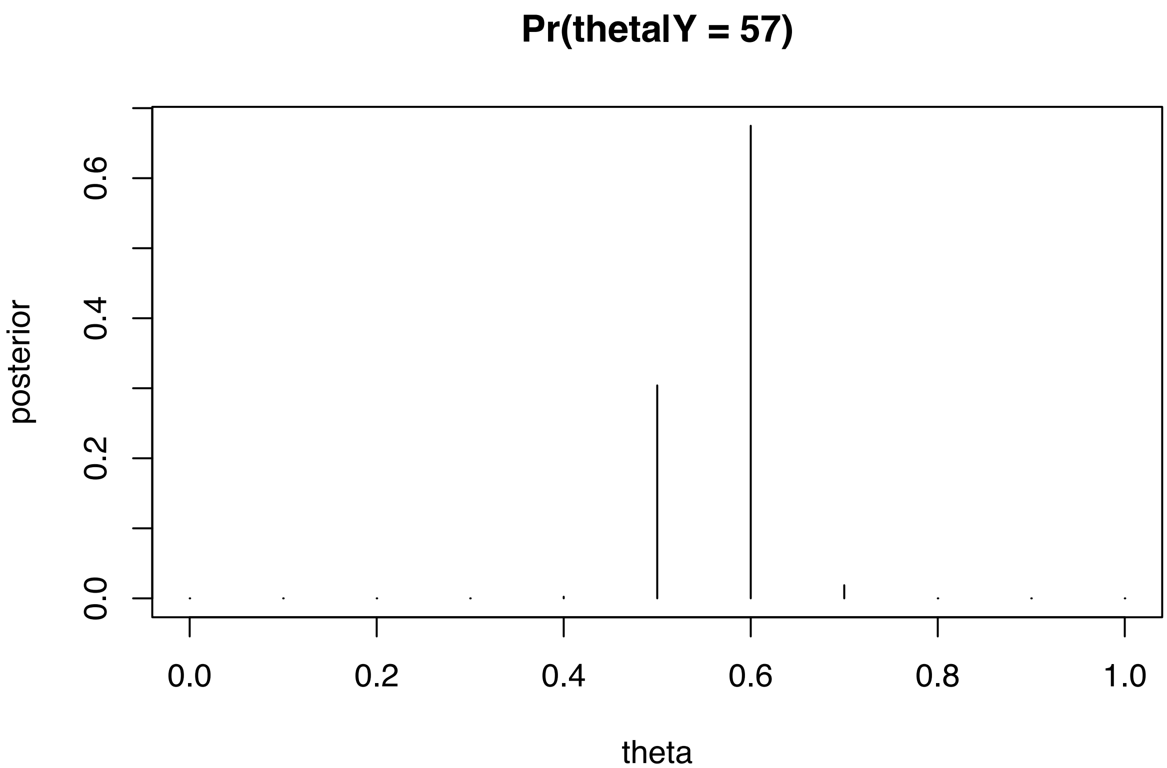
```
theta = seq(0, 1, 0.1)
prob = choose(100, 57) * (theta^57) * (1 - theta)^(100 - 57)
plot(theta, prob, type = 'h', main = 'Pr(Y = 57|theta)')
```



c)

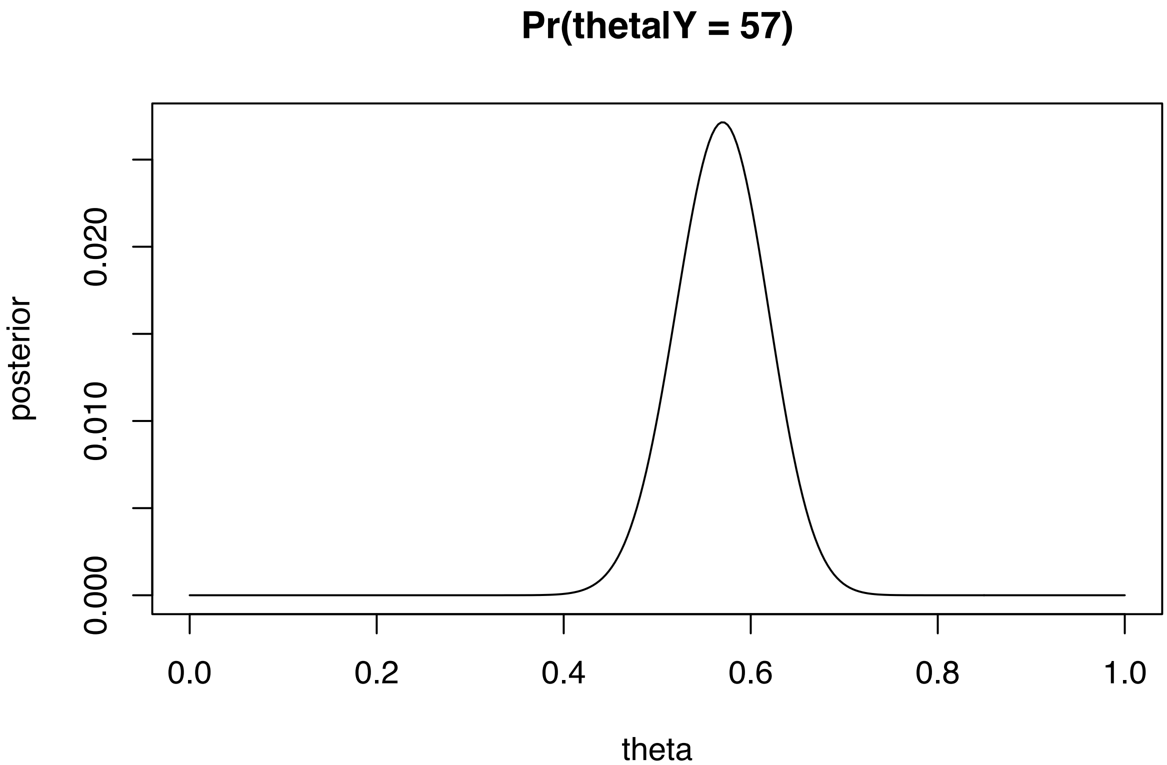
```
prior = 1/11
theta = seq(0, 1, 0.1)
likelihood = choose(100, 57) * (theta^57) * (1 - theta)^(100 - 57)
```

```r
posterior = likelihood * prior
posterior = posterior/sum(posterior)
plot(theta, posterior, type = 'h', main = 'Pr(theta|Y = 57)')
```



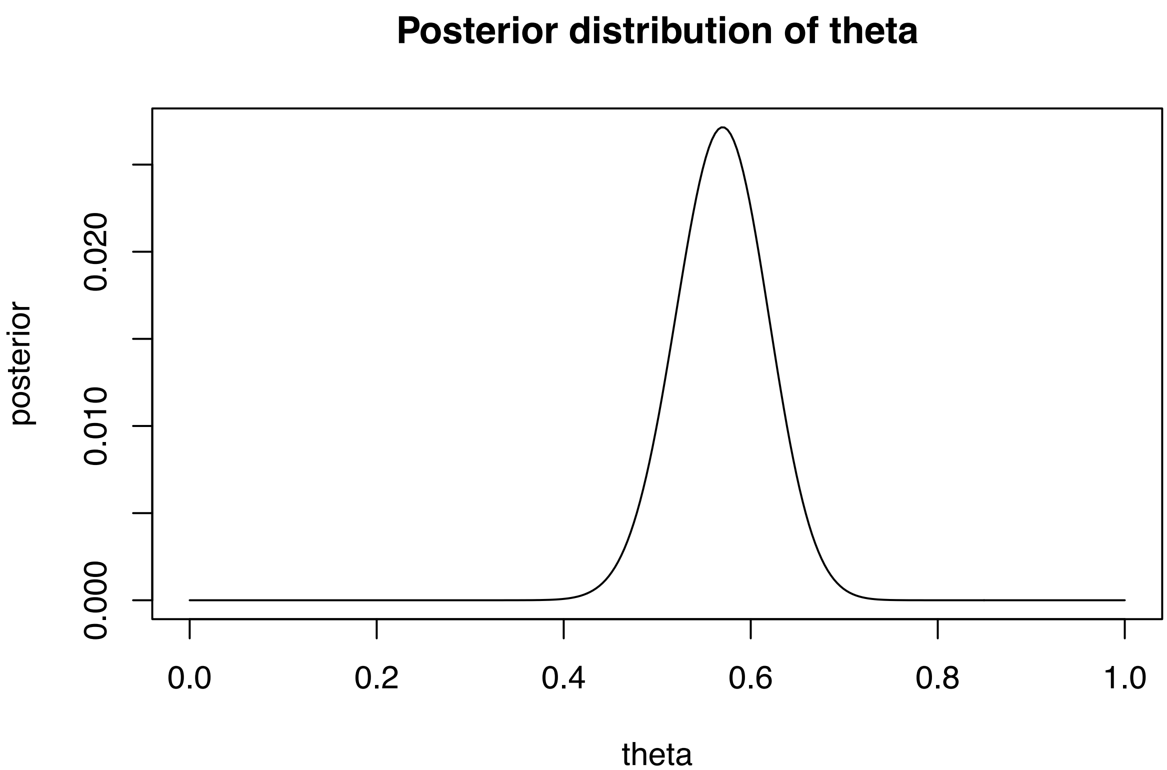
d)

```r
prior = 1
theta = seq(0, 1, length = 300)
likelihood = choose(100, 57) * (theta^57) * (1 - theta)^(100 - 57)
posterior = likelihood * prior
posterior = posterior/sum(posterior)
plot(theta, posterior, type = 'l', main = 'Pr(theta|Y = 57)')
```



e)

```
theta = seq(0, 1, length = 300)
posterior = dbeta(theta, 1 + 57, 1 + 100 - 57)
posterior = posterior/sum(posterior)
plot(theta, posterior, type = 'l', main = 'Posterior distribution of theta')
```



The plot from the exact Beta Distribution is almost identical with the plot drawn from (d), where the posterior mode being a value slightly smaller that 0.6. (c) with discrete prior did a pretty good job as well but the posterior estimate for theta was exactly 0.6 in (c) where we found out that the actual value must be slightly smaller that 0.6. Also by looking at (b) the result seems quiet reasonable.

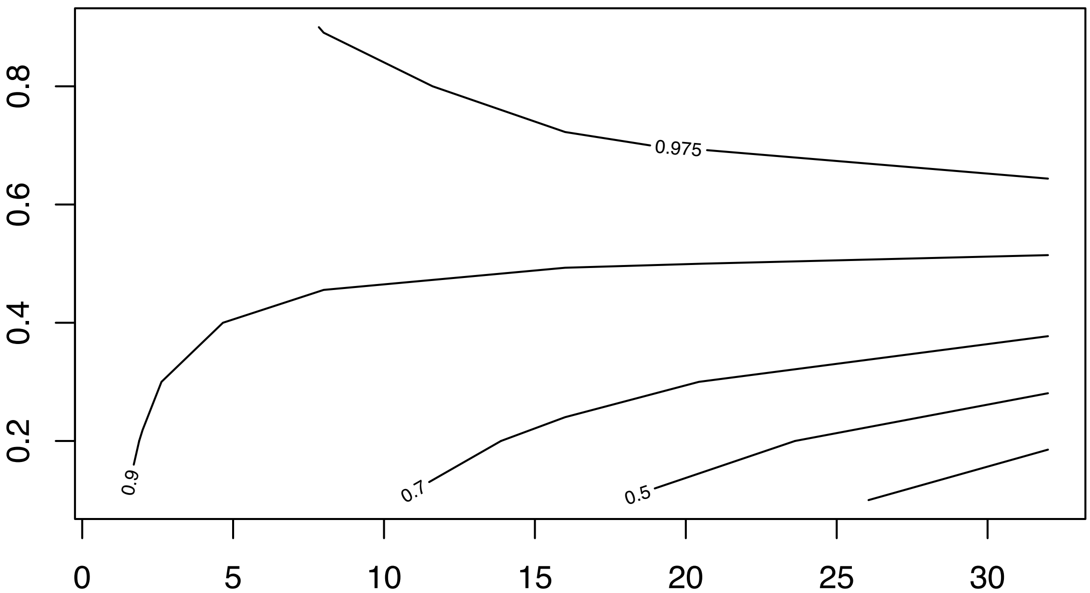
1-(b) # 3.2

```
theta_0 = seq(0.1, 0.9, 0.1)
n_0 = c(1, 2, 8, 16, 32)

theta_0 = rep(theta_0, each = 5)
n_0 = rep(n_0, times = 9)
a = theta_0 * n_0
b = (1 - theta_0) * n_0
df = data.frame(theta_0, n_0, a, b)
print(df)
```

```
##    theta_0 n_0   a    b
## 1      0.1   1 0.1  0.9
## 2      0.1   2 0.2  1.8
## 3      0.1   8 0.8  7.2
## 4      0.1  16 1.6 14.4
## 5      0.1  32 3.2 28.8
## 6      0.2   1 0.2  0.8
## 7      0.2   2 0.4  1.6
## 8      0.2   8 1.6  6.4
## 9      0.2  16 3.2 12.8
```

```
## 10     0.2  32  6.4 25.6
## 11     0.3   1  0.3  0.7
## 12     0.3   2  0.6  1.4
## 13     0.3   8  2.4  5.6
## 14     0.3  16  4.8 11.2
## 15     0.3  32  9.6 22.4
## 16     0.4   1  0.4  0.6
## 17     0.4   2  0.8  1.2
## 18     0.4   8  3.2  4.8
## 19     0.4  16  6.4  9.6
## 20     0.4  32 12.8 19.2
## 21     0.5   1  0.5  0.5
## 22     0.5   2  1.0  1.0
## 23     0.5   8  4.0  4.0
## 24     0.5  16  8.0  8.0
## 25     0.5  32 16.0 16.0
## 26     0.6   1  0.6  0.4
## 27     0.6   2  1.2  0.8
## 28     0.6   8  4.8  3.2
## 29     0.6  16  9.6  6.4
## 30     0.6  32 19.2 12.8
## 31     0.7   1  0.7  0.3
## 32     0.7   2  1.4  0.6
## 33     0.7   8  5.6  2.4
## 34     0.7  16 11.2  4.8
## 35     0.7  32 22.4  9.6
## 36     0.8   1  0.8  0.2
## 37     0.8   2  1.6  0.4
## 38     0.8   8  6.4  1.6
## 39     0.8  16 12.8  3.2
## 40     0.8  32 25.6  6.4
## 41     0.9   1  0.9  0.1
## 42     0.9   2  1.8  0.2
## 43     0.9   8  7.2  0.8
## 44     0.9  16 14.4  1.6
## 45     0.9  32 28.8  3.2
```

```r
n = 100; x = 57
pr = numeric()
for (i in 1:nrow(df)){
  pr[i] = 1 - pbeta(0.5, x + df$a[i], n - x + df$b[i])
}

theta_0 = seq(0.1, 0.9, 0.1)
n_0 = c(1, 2, 8, 16, 32)
pr = matrix(pr, nrow = length(theta_0), byrow = T)
contour(n_0, theta_0, t(pr), levels=c(0.1,0.3,0.5,0.7,0.9,0.975))
```

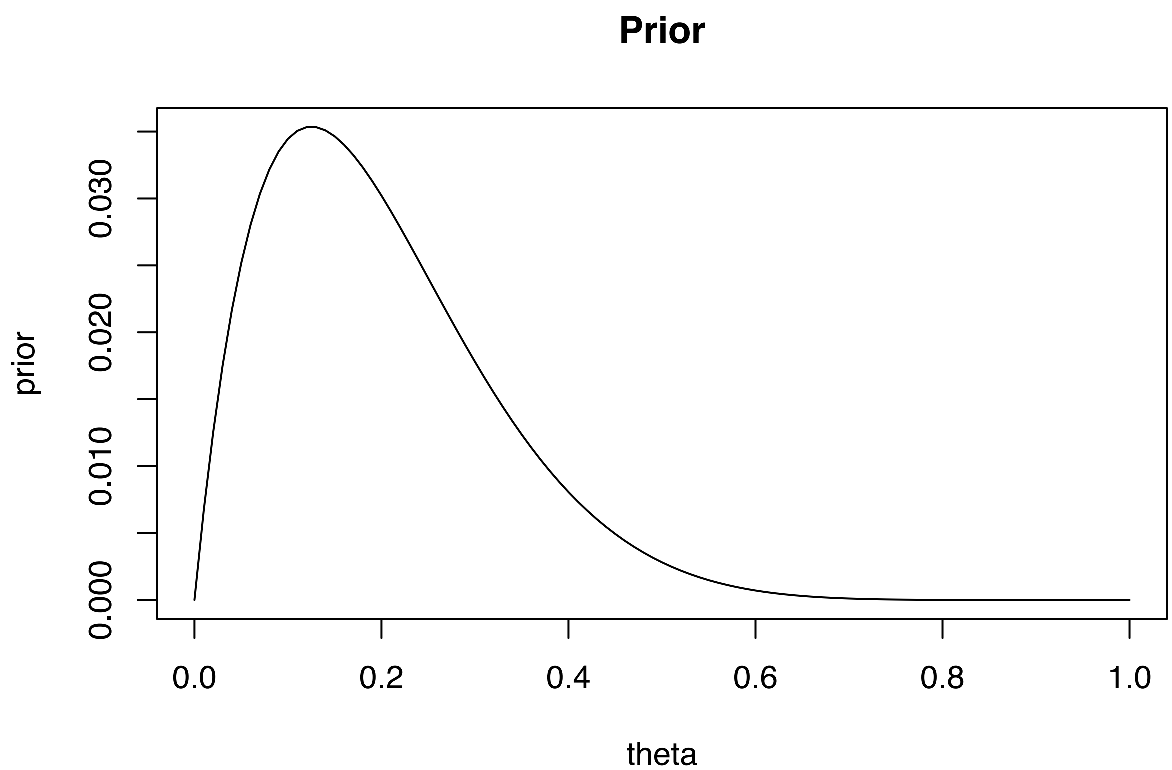


This plot can be used to explain to someone to believe that theta is over 0.5 based on the given data, we can see that even with lower prior belief are generally 90% or more certain that the theta is over 0.5. Those with high prior belief are very highly certain(97.5%) that theta is over 0.5.

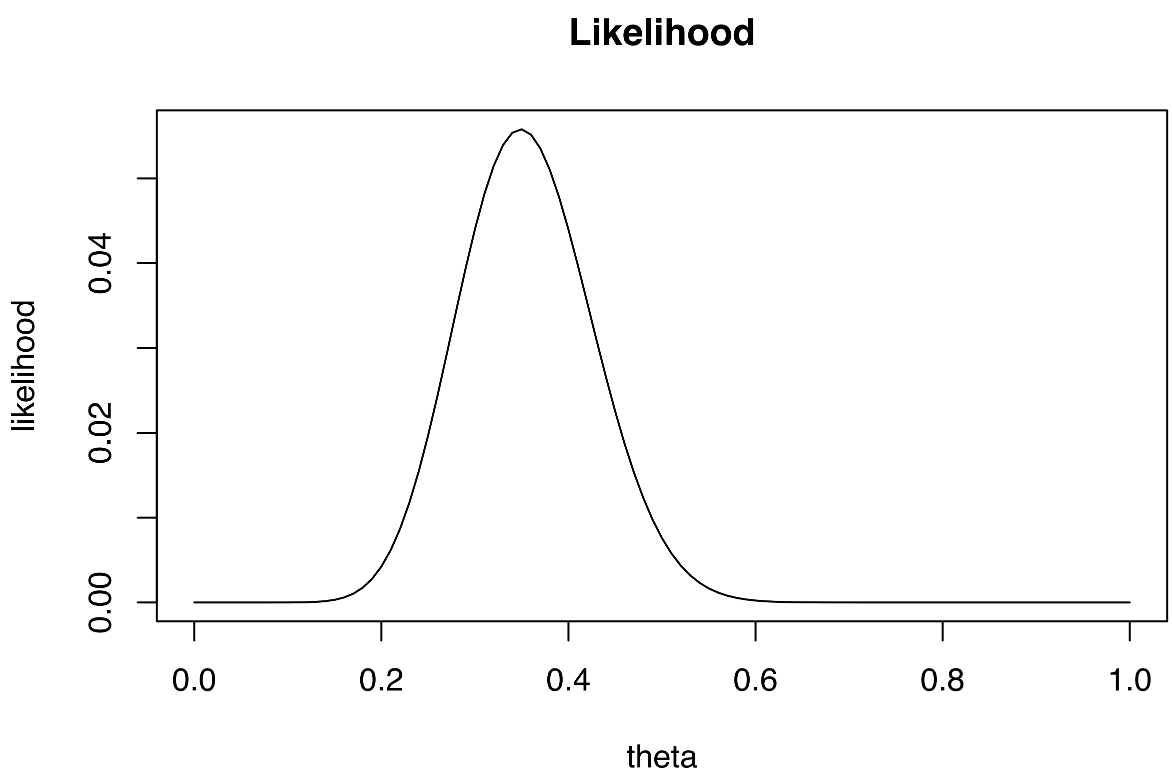
1-(c) # 3.4

a)

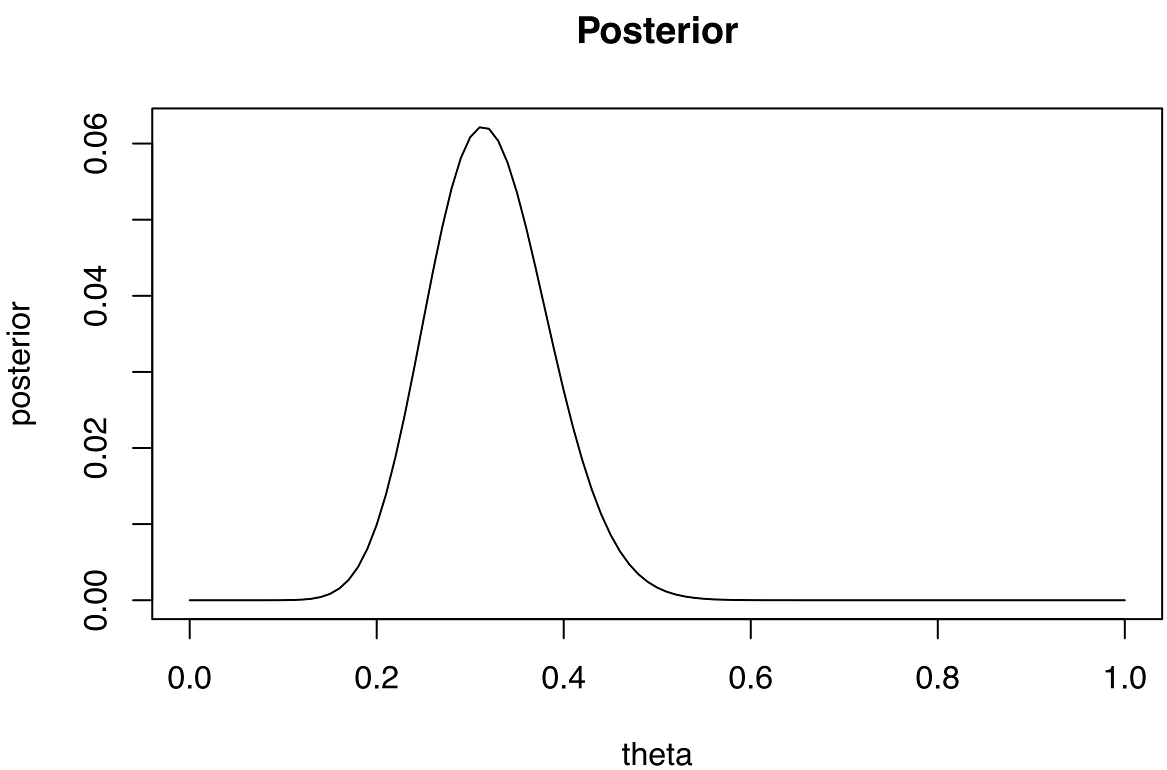
```
n = 43; y = 15
a = 2; b = 8
theta = seq(0, 1, 0.01)
prior = dbeta(theta, a, b); prior = prior/sum(prior)
likelihood = dbinom(y, n, theta); likelihood = likelihood/sum(likelihood)
posterior = dbeta(theta, a + y, b + n - y); posterior = posterior/sum(posterior)
plot(theta, prior, type = 'l', main = 'Prior')
```




```
plot(theta, likelihood, type = 'l', main = 'Likelihood')
```




```
plot(theta, posterior, type  = 'l', main = 'Posterior')
```




```
cat("posterior mean of theta is", (a + y)/(a + b + n))
```

```
## posterior mean of theta is 0.3207547
```

```
cat("\nposterior mode of theta is",
theta[which.max(posterior)])
```

```
##
## posterior mode of theta is 0.31
```

```
cat("\nposterior sd of theta is", sqrt(((a + y) * (b + n - y))/(((a + b + n)^2) * (a + b+ n + 1))))
```

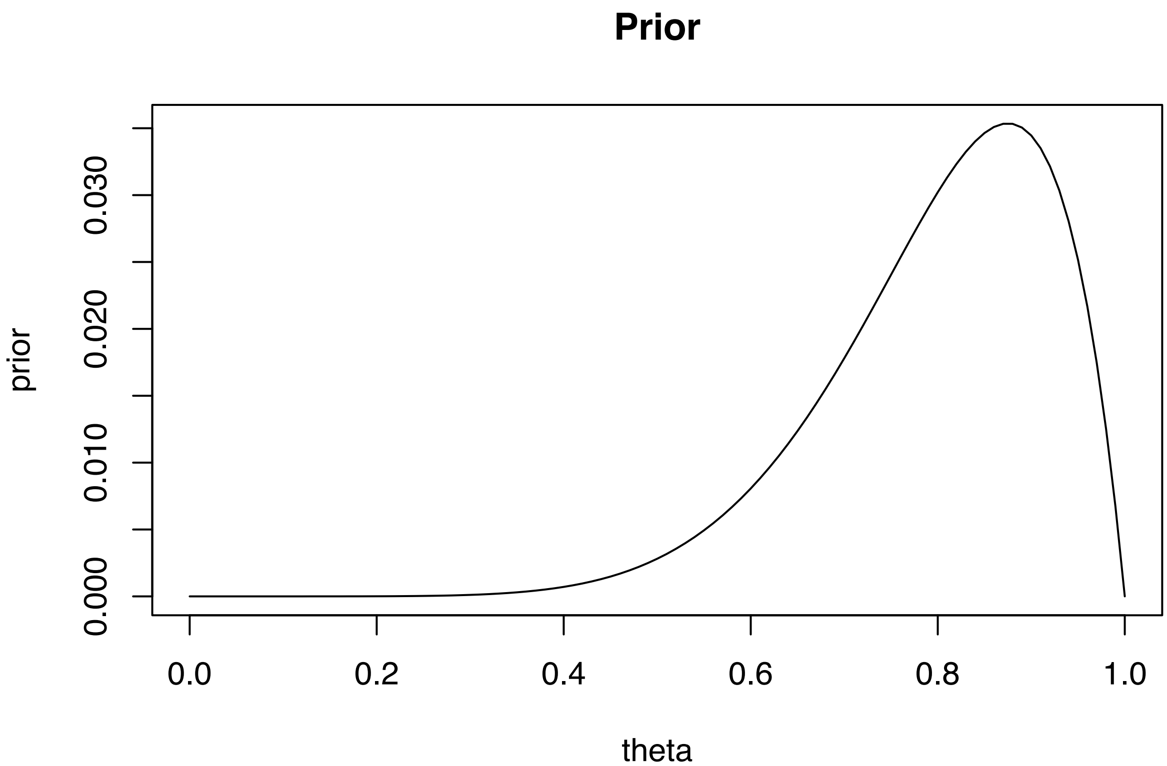
```
##
## posterior sd of theta is 0.0635189
```

```
cat("\n95% quantile-based confidence interval is", qbeta(c(0.025, 0.975), a + y, b + n - y))
```

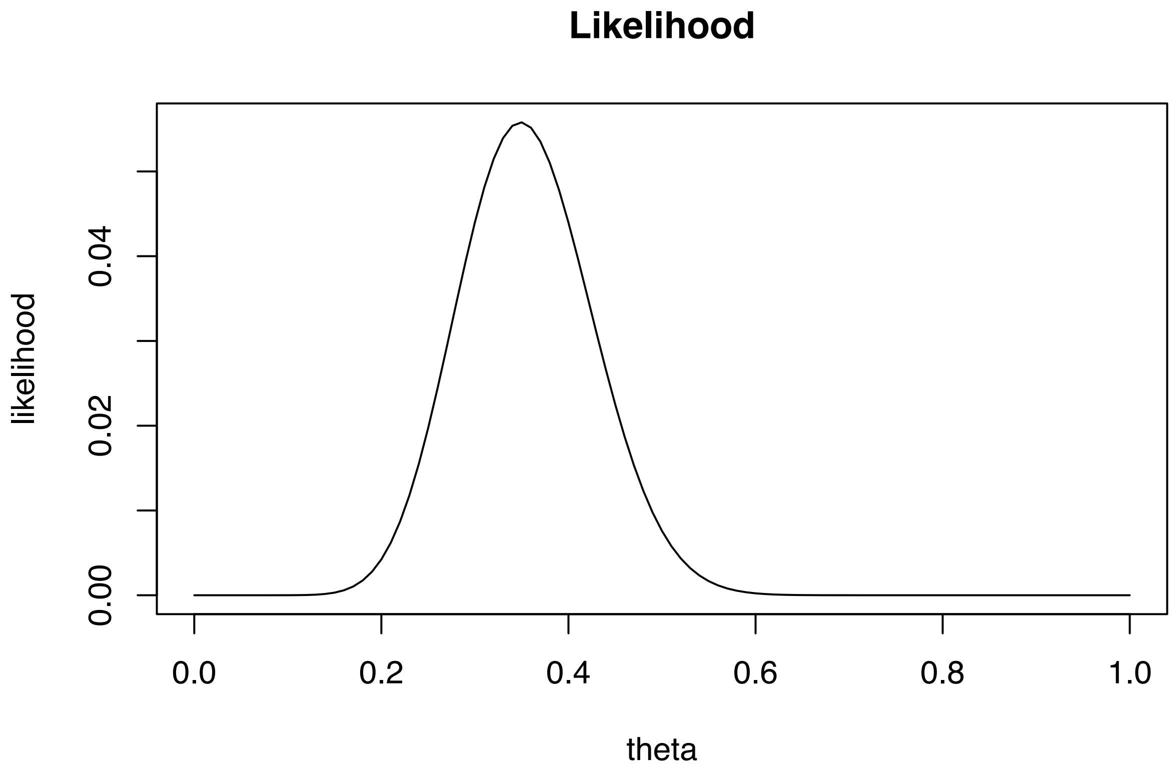
```
##
## 95% quantile-based confidence interval is 0.2032978 0.451024
```

b)

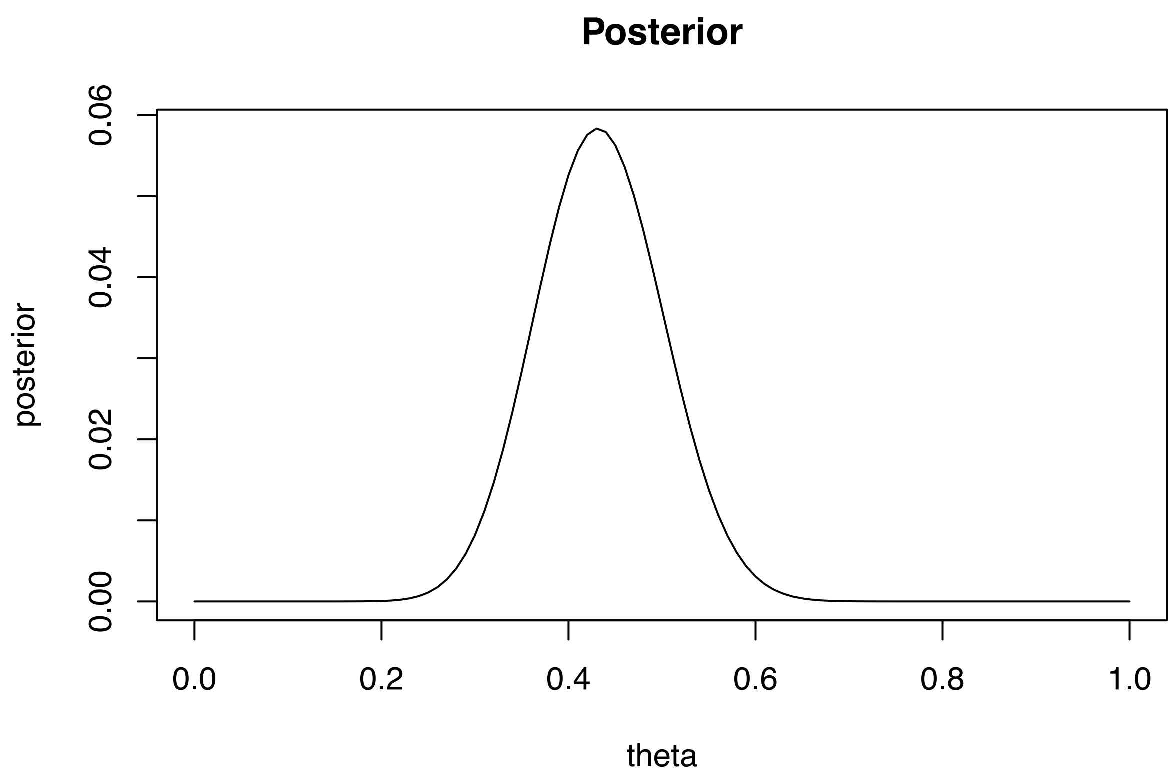
```r
n = 43; y = 15
a = 8; b = 2
theta = seq(0, 1, 0.01)
prior = dbeta(theta, a, b); prior = prior/sum(prior)
likelihood = dbinom(y, n, theta); likelihood = likelihood/sum(likelihood)
posterior = dbeta(theta, a + y, b + n - y); posterior = posterior/sum(posterior)
plot(theta, prior, type = 'l', main = 'Prior')
```




```r
plot(theta, likelihood, type = 'l', main = 'Likelihood')
```




```r
plot(theta, posterior, type  = 'l', main = 'Posterior')
```




```r
cat("posterior mean of theta is", (a + y)/(a + b + n))
```

```
## posterior mean of theta is 0.4339623
```

```r
cat("\nposterior mode of theta is",
theta[which.max(posterior)])
```

```
##
## posterior mode of theta is 0.43
```

```r
cat("\nposterior sd of theta is", sqrt(((a + y) * (b + n - y))/(((a + b + n)^2) * (a + b+ n + 1))))
```

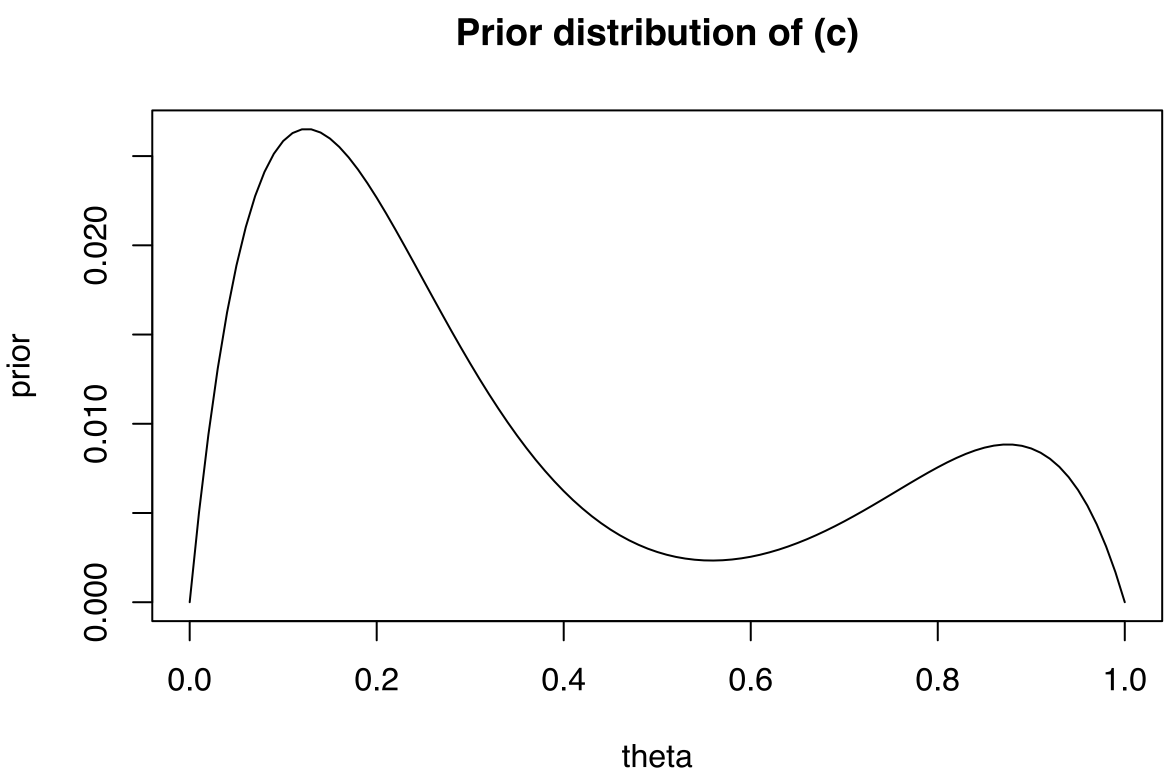
```
##
## posterior sd of theta is 0.06744532
```

```r
cat("\n95% quantile-based confidence interval is", qbeta(c(0.025, 0.975), a + y, b + n - y))
```

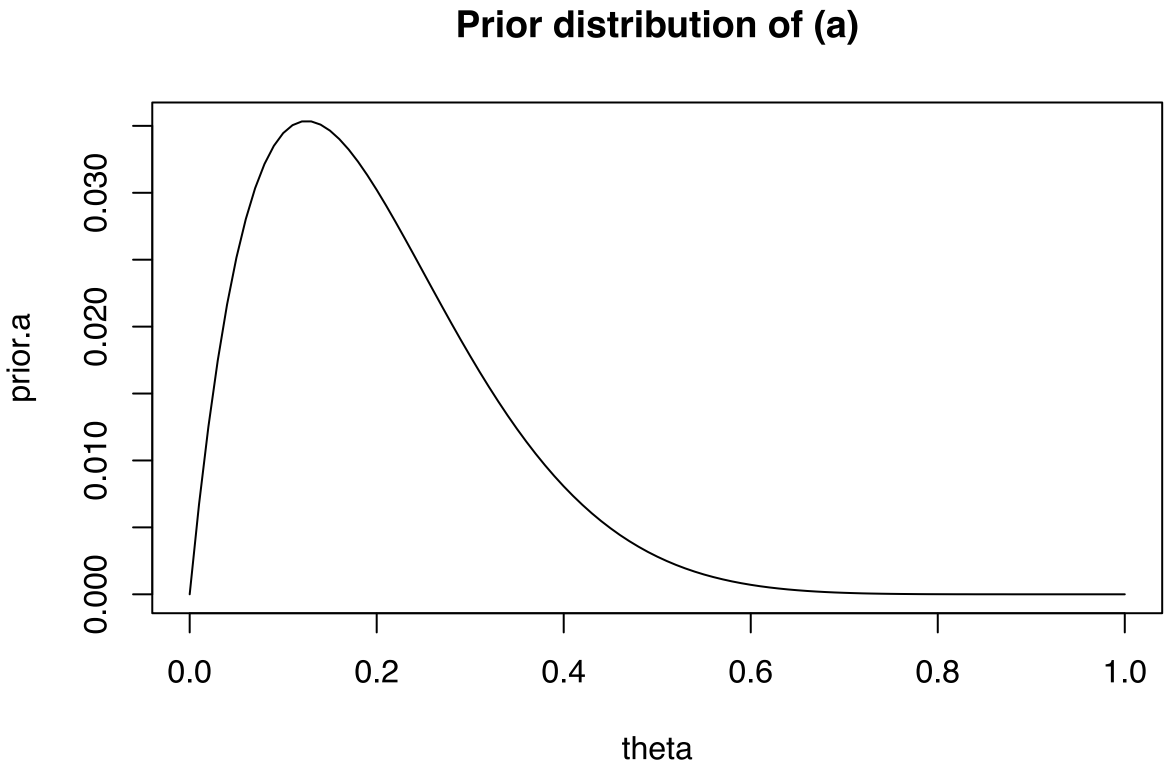
```
##
## 95% quantile-based confidence interval is 0.3046956 0.5679528
```

c)

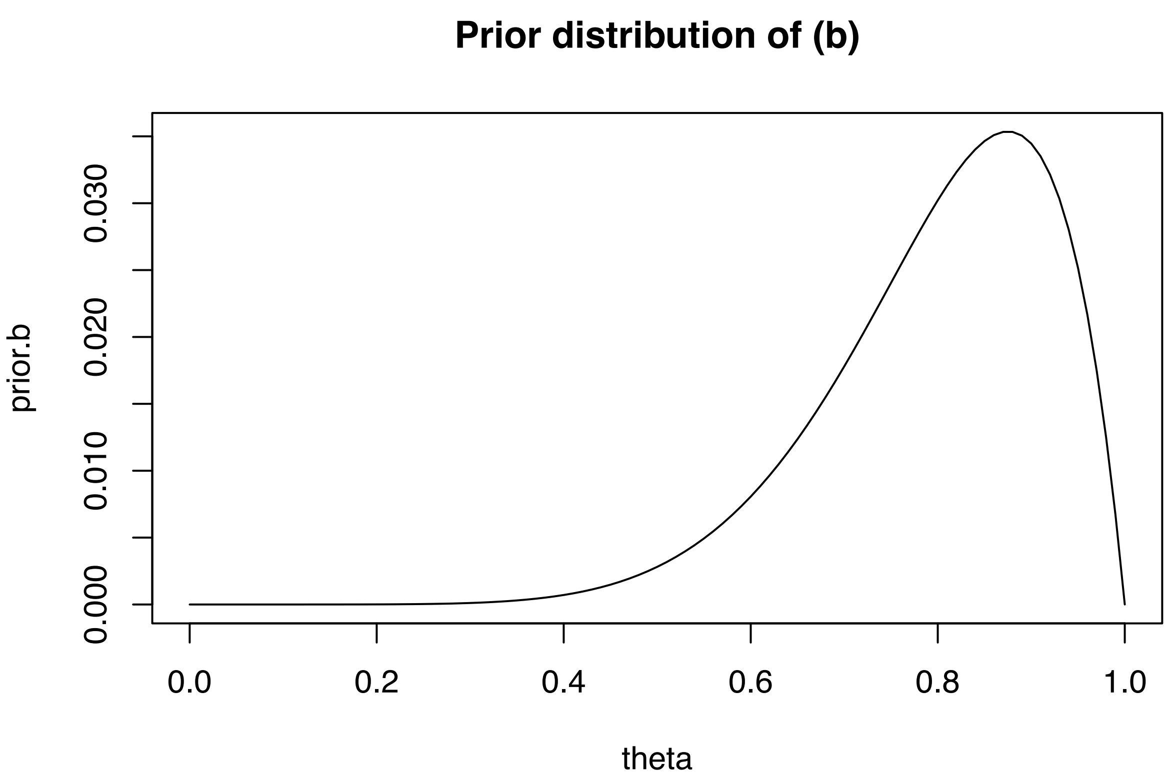
```r
theta = seq(0, 1, 0.01)
prior = (1/4) * (gamma(10)/(gamma(2) * gamma(8))) * (3 * theta * (1 - theta)^7 + theta^7 * (1 - theta))
prior = prior/sum(prior)
prior.a = dbeta(theta, 2, 8); prior.b = dbeta(theta, 8, 2)
prior.a = prior.a/sum(prior.a); prior.b = prior.b/sum(prior.b)
plot(theta, prior, type = 'l', main = 'Prior distribution of (c)')
```



```r
plot(theta, prior.a, type = 'l', main = 'Prior distribution of (a)')
```




```r
plot(theta, prior.b, type = 'l', main = 'Prior distribution of (b)')
```

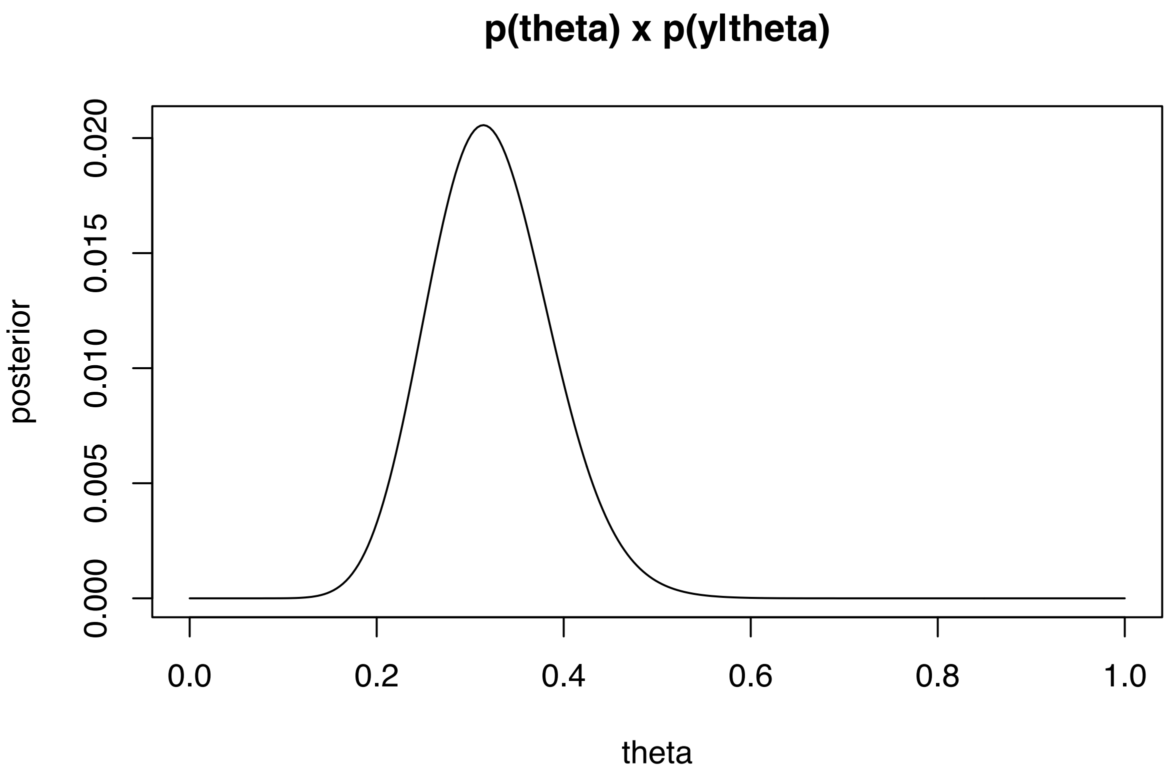


The distribution has two local maximum point unlike the prior distributions from a) or b). And the two points are identical with the points from a) and b) which is because it is the mixture of a beta(2, 8) and beta(8, 2) prior distribution as it is described. The prior opinion maybe that the probability that a particular event will occur is either high or low, but it is more likely to be low.

d)

iii.

```
theta = seq(0, 1, length = 300)
posterior = 18 * choose(43, 15) * (3 * (theta^16) * ((1 - theta)^35) + (theta^22) * ((1 - theta)^29))
posterior = posterior/sum(posterior)
plot(theta, posterior, type = 'l', main = 'p(theta) x p(y|theta)')
```




```
cat("\nposterior mode of theta is",
theta[which.max(posterior)])
```

```
##
## posterior mode of theta is 0.3143813
```

The posterior mode is between the posterior mode of (a) = 0.31 and (b) = 0.43 but is much closer to the posterior mode of (a).

d) i. $p(\theta) \times p(y|\theta) = \frac{1}{4}\frac{\Gamma(10)}{\Gamma(2)\Gamma(8)}[3\theta(1-\theta)^7 + \theta^7(1-\theta)] \times \binom{43}{15}\theta^{15}(1-\theta)^{(43-15)}$

$$= 18\binom{43}{15}\theta^{16}(1-\theta)^{29}[3(1-\theta)^6 + \theta^6]$$

ii. $18\binom{43}{15}\theta^{16}(1-\theta)^{29}[3(1-\theta)^6+\theta^6] = 18\binom{43}{15}[3\theta^{16}(1-\theta)^{35} + \theta^{22}(1-\theta)^{29}]$

The posterior distribution is the mixture of Beta(17,36) and Beta(23,30) which are the posteriors from (a) and (b).

e) $P(\theta|y) = \frac{P(\theta)P(y|\theta)}{\int P(\theta)P(y|\theta)d\theta}$

let $P(\theta) = \alpha f_1(\theta) + (1-\alpha)f_2(\theta)$, then,

$P(\theta|y) = \frac{P(y|\theta)\alpha f_1(\theta) + P(y|\theta)(1-\alpha)f_2(\theta)}{\int[P(y|\theta)\alpha f_1(\theta) + P(y|\theta)(1-\alpha)f_2(\theta)]d\theta}$, let $g_i(y) = \int[P(y|\theta)f_i(\theta)]d\theta$

$= \frac{\frac{P(y|\theta)f_1(\theta)}{g_1(y)}\alpha g_1(y) + \frac{P(y|\theta)f_2(\theta)}{g_2(y)}(1-\alpha)g_2(y)}{\alpha g_1(y) + (1-\alpha)g_2(y)}$, let $W_1 = \frac{\alpha g_1(y)}{\alpha g_1(y)+(1-\alpha)g_2(y)}$, $W_2 = \frac{(1-\alpha)g_2(y)}{\alpha g_1(y)+(1-\alpha)g_2(y)} = (1-W_1)$

$= \frac{P(y|\theta)f_1(\theta)}{g_1(y)}W_1 + \frac{P(y|\theta)f_2(\theta)}{g_2(y)}W_2$

$= \frac{P(y|\theta)f_1(\theta)}{\int[P(y|\theta)f_1(\theta)]d\theta}W_1 + \frac{P(y|\theta)f_2(\theta)}{\int[P(y|\theta)f_2(\theta)]d\theta}W_2$

$= P_1(\theta|y)W_1 + P_2(\theta|y)W_2$

Therefore, the general formula for the weights of the mixture distribution is $W_1 = \frac{\alpha g_1(y)}{\alpha g_1(y)+(1-\alpha)g_2(y)}$, $W_2 = (1-W_1)$ where $g_i(y) = \int[P(y|\theta)f_i(\theta)]d\theta$. The posterior weight of a component of mixture distribution is the marginal density of y for the component over the sum of marginal density of each component.

1.(e) #3.10

a) $\theta \sim Beta(a,b)$, $P_\theta(\theta) = \frac{1}{B(a,b)}\theta^{a-1}(1-\theta)^{b-1}$

Because $\psi = \log(\frac{\theta}{1-\theta})$,

$$\frac{\theta}{1-\theta} = e^{\psi}$$

$$\frac{1-\theta}{\theta} = \frac{1}{\theta} - 1 = e^{-\psi}$$

$$\frac{1}{\theta} = e^{-\psi}+1$$

$$\theta = \frac{1}{1+e^{-\psi}} = h(\psi)$$

$\frac{dh}{d\psi} = \frac{e^{-\psi}}{(1+e^{-\psi})^2}$

$P_\psi(\psi) = P_\theta(h(\psi)) \times \left|\frac{dh}{d\psi}\right|$

$= \frac{1}{B(a,b)}\left(\frac{1}{1+e^{-\psi}}\right)^{a-1}\left(\frac{e^{-\psi}}{1+e^{-\psi}}\right)^{b-1}\frac{e^{-\psi}}{(1+e^{-\psi})^2}$

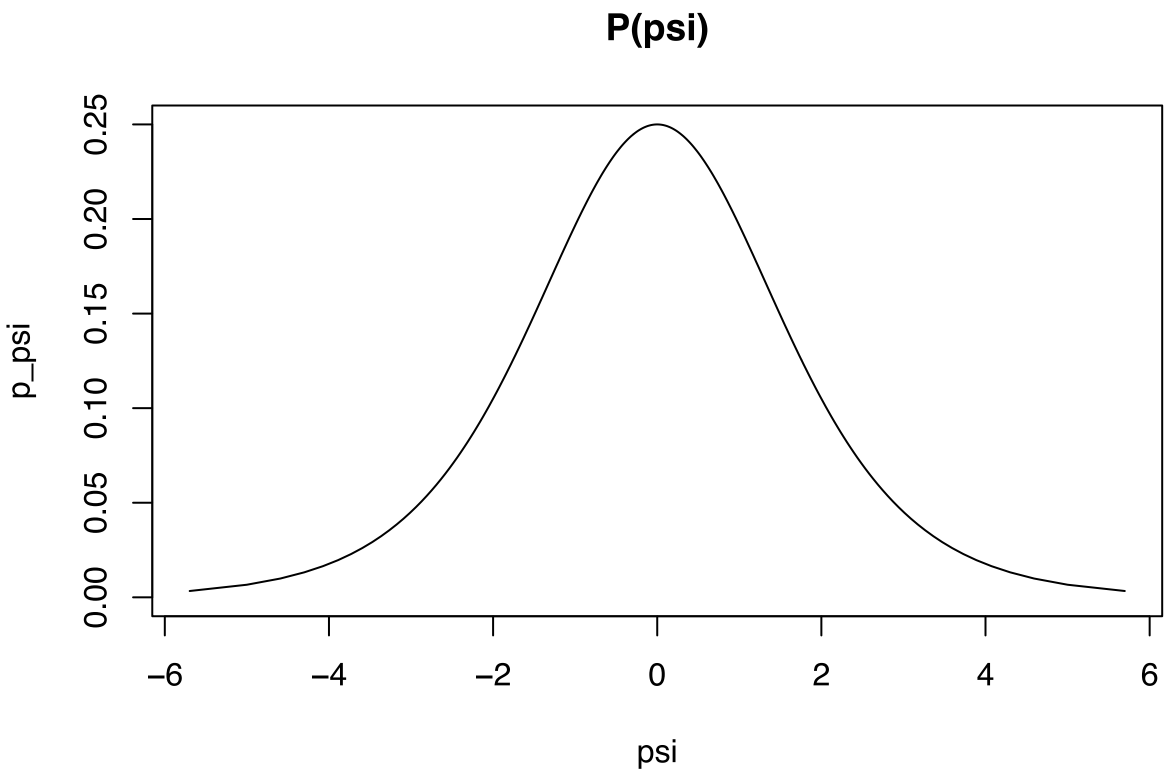
$= \frac{1}{B(a,b)}\frac{e^{-b\psi}}{(1+e^{-\psi})^{a+b}}$

and when $a=b=1$

1-(e) # 3.10

a)

```
theta = seq(0, 1, length = 300)
psi = log(theta/(1 - theta))
p_psi = (1/beta(1, 1)) * exp(-psi)/(1 + exp(-psi))^2
plot(psi, p_psi, type = 'l', main = 'P(psi)')
```



2

(a)

```
set.seed(1); n = 10^4

pmf = c(0.45, 0.55)

rng.discrete = function(pmf){
  repeat{y = rbinom(1, 1, prob = 0.5) ; v = runif(1)
if (v < pmf[y + 1]/max(pmf))
break}
return(floor(y))}

rng.Y = replicate(n, rng.discrete(pmf))
table(rng.Y)/n
```

```
## rng.Y
##      0      1
## 0.4562 0.5438
```

The empirical marginal p.m.f of Y based on random numbers are 0.4562 for Y = 0 and 0.5438 for Y = 1, where the exact marginal p.m.f of Y are 0.45 for Y = 0 and 0.5 for Y = 1. It is pretty well matched.

(b)

```r
set.seed(1); n = 10^4

n = 10^4
jointpmf = matrix(c(0.15, 0.15, 0.15, 0.15, 0.2, 0.2), 2, 3, byrow = T)
dimnames(jointpmf)[[1]] = 0:1; dimnames(jointpmf)[[2]] = 0:2
cat("Joint Pmf\n")
```

```
## Joint Pmf
```

```r
jointpmf
```

```
##      0    1    2
## 0 0.15 0.15 0.15
## 1 0.15 0.20 0.20
```

```r
cat("\n")
```

```r
rng.comp = function(jointp, nsim){
 jointsamp = matrix(0, nsim, 2);
  for (k in 1:nsim){
    ypmf = apply(jointp, 1, sum) #marginal_dist P(Y)
    y = sample(0:1, 1, prob = ypmf, replace = T)
    x = sample(0:2, 1, prob = jointp[(y + 1), ]/ypmf[(y + 1)], replace = T)# P(X, Y)
    jointsamp[k,] = c(x, y)}
  return(jointsamp)}


xy.rng = data.frame(rng.comp(jointpmf, n))
names(xy.rng) = c("X", "Y"); table(xy.rng)/n
```

```
##    Y
## X        0      1
##   0 0.1511 0.1520
##   1 0.1476 0.2004
##   2 0.1454 0.2035
```

```r
cat("\nP(X = x)\n")
```

```
##
## P(X = x)
```

```r
apply(table(xy.rng)/n, 1, sum)
```

```
##      0      1      2
## 0.3031 0.3480 0.3489
```

The empirical marginal p.m.f of X based on random numbers are 0.3031 for X = 0, 0.3480 for X = 1 and 0.3489 for X = 2. The actual values of P(X = x) are 0.3, 0.35, 0.35 for each. It is pretty well matched.

(c)

```r
set.seed(1); n = 10^4

trans.mat = function(A) {
  n = length(A[,1]); temp = A
  for (i in 1:n){temp[i,] = A[i,]/sum(A[i,])}
  return(temp)
}

gibbs_discrete = function(pmat1, pmat2, i = 1, iter){
  jointsamp = matrix(0, iter, 2)
  for (k in 1:iter){
    j = sample(0:2, 1, prob = pmat1[(i + 1),])
    i = sample(0:1, 1, prob = pmat2[(j + 1),])
    jointsamp[k,] = c(i, j)}
  return(jointsamp)
}


Pmat.YX = trans.mat(jointpmf); Pmat.XY = trans.mat(t(jointpmf))
nburnin = 2000; niter = nburnin + n
gibbs_samp = data.frame(gibbs_discrete(Pmat.YX, Pmat.XY, 1, niter))
names(gibbs_samp) = c("Y", "X")
X.Y = table(gibbs_samp[(nburnin+1):(niter), ])/n
colnames(X.Y) = c(0, 1, 2)
rownames(X.Y) = c(0, 1)
cat('P(X=x, Y = y)\n')
```

```
## P(X=x, Y = y)
```

```r
X.Y
```

```
##    X
## Y        0      1      2
##   0 0.1498 0.1533 0.1534
##   1 0.1457 0.2001 0.1977
```

```r
powermat = function(mat, k){
  if (k == 0) return (diag(dim(mat)[1]))
  if (k == 1) return(mat)
```

```
  if (k > 1)  return(mat %*% powermat(mat, k-1))
}


Pmat.Y <- Pmat.YX %*% Pmat.XY
Pmat.X <- Pmat.XY %*% Pmat.YX

cat('\nP(X=x)\n')
```

```
##
## P(X=x)
```

```
powermat(Pmat.X, 100)
```

```
##     0    1    2
## 0 0.3 0.35 0.35
## 1 0.3 0.35 0.35
## 2 0.3 0.35 0.35
```

```
cat('\nP(Y=y)\n')
```

```
##
## P(Y=y)
```

```
powermat(Pmat.Y, 100)
```

```
##      0    1
## 0 0.45 0.55
## 1 0.45 0.55
```

The estimates for P(X = x, Y = y) are 0.1498 for X = 0, Y = 0, 0.1533 for X = 1, Y = 0, 0.1534 for X = 2, Y = 0, 0.1457 for X = 0, Y = 1, 0.2001 for X = 1, Y = 1 and 0.1977 for X = 2, Y = 1. The actual values are 0.15, 0.15, 0.15, 0.15, 0.20, 0.20 for each. It is pretty well matched. The estimates for P(X = x) are 0.30 for X = 0, 0.35 for X = 1 and 0.35 for X = 2. The actual values of P(X = x) are 0.3, 0.35, 0.35 for each. It is exactly the same. The estimates for P(Y = y) are 0.45 for Y = 0, 0.55 for Y = 1. The actual values of P(Y = y) are 0.45, 0.55 for each. It is also exactly the same.

3

3.(a) $P_A(\theta|x) \propto P_A(x|\theta)P_A(\theta) \propto \theta^{710}(1-\theta)^{290}\,\theta = \theta^{711}(1-\theta)^{290}$

$P_B(\theta|x) \propto P_B(x|\theta)P_B(\theta) \propto \theta^{710}(1-\theta)^{290}\,\theta^2 = \theta^{712}(1-\theta)^{290}$

(b)

Posterior distribution of statistician A follows Beta(712,291). Posterior distribution of statistician B follows Beta(713,291).

```
set.seed(42)
n = 1000; x = 710
theta = seq(0, 1, length = n)
posterior.A = dbeta(theta, 712, 291)
posterior.A = posterior.A/sum(posterior.A)
cat("Using grid approximation,\n")
```

```
## Using grid approximation,
```

```r
sum(posterior.A[theta < 0.4])
```

```
## [1] 4.29697e-88
```

```r
cat('Using Monte Carlo Approximation,\n')
```

```
## Using Monte Carlo Approximation,
```

```r
posterior.B = rbeta(1000, 712, 291)
mean(posterior.B < 0.4)
```

```
## [1] 0
```

```r
cat("\n Exact value using pbeta()\n")
```

```
##
##  Exact value using pbeta()
```

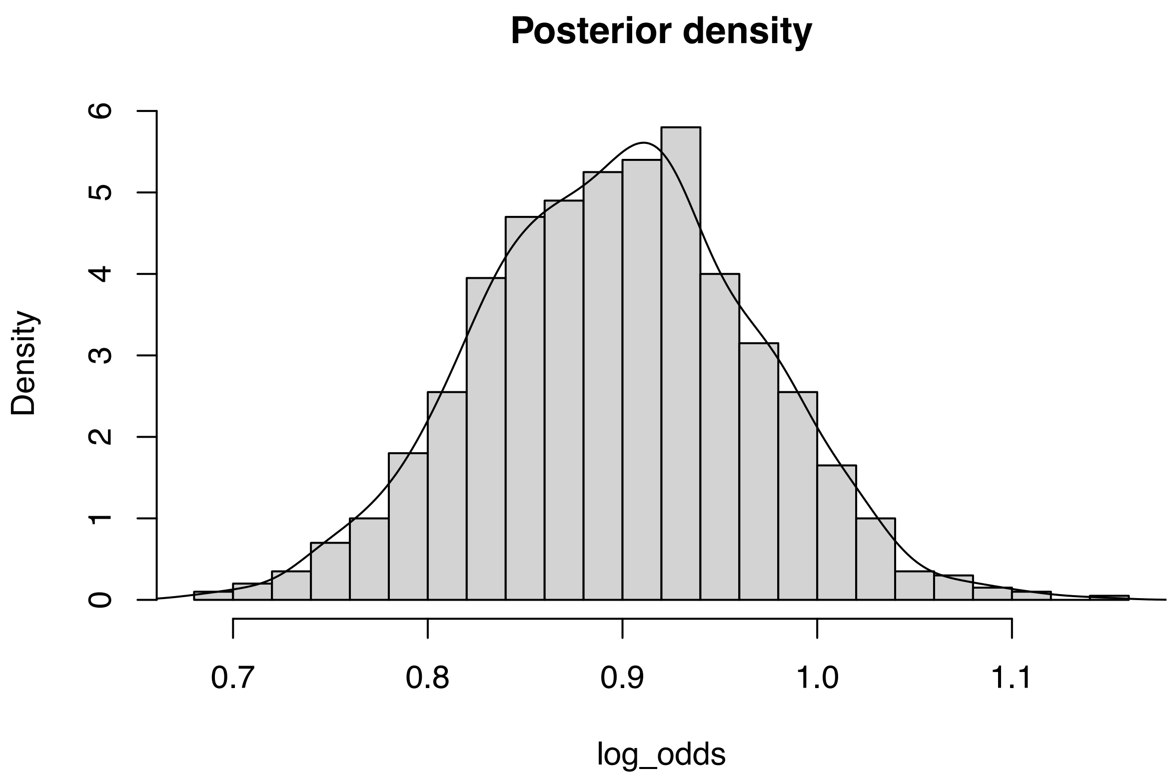
```r
pbeta(0.4, 712, 291)
```

```
## [1] 5.244442e-88
```

The probability that theta is less than 0.4 based on the given prior pdf are all zero(approximately) regardless of the method used.

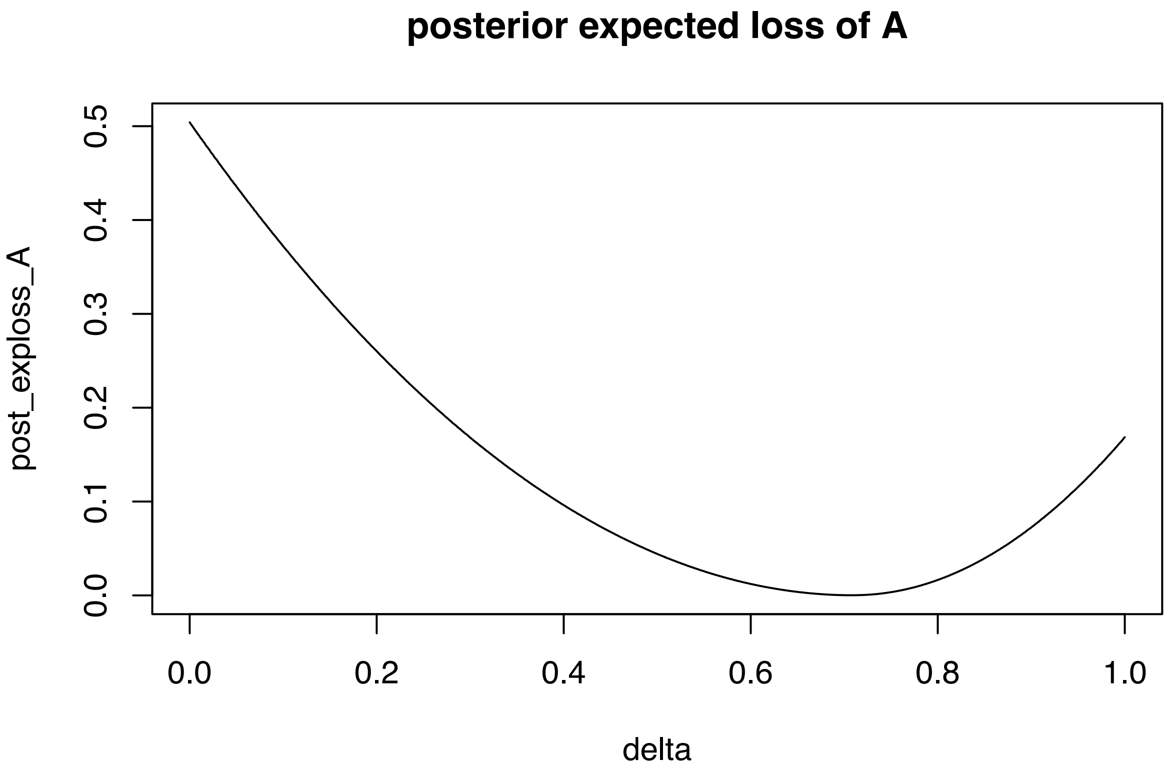
(c)

```r
n = 1000; set.seed(1)
thetasamp = rbeta(n, 713, 291)
log_odds = log(thetasamp / (1-thetasamp))
hist(log_odds, breaks=25, main="Posterior density",freq=F)
lines(density(log_odds))
```

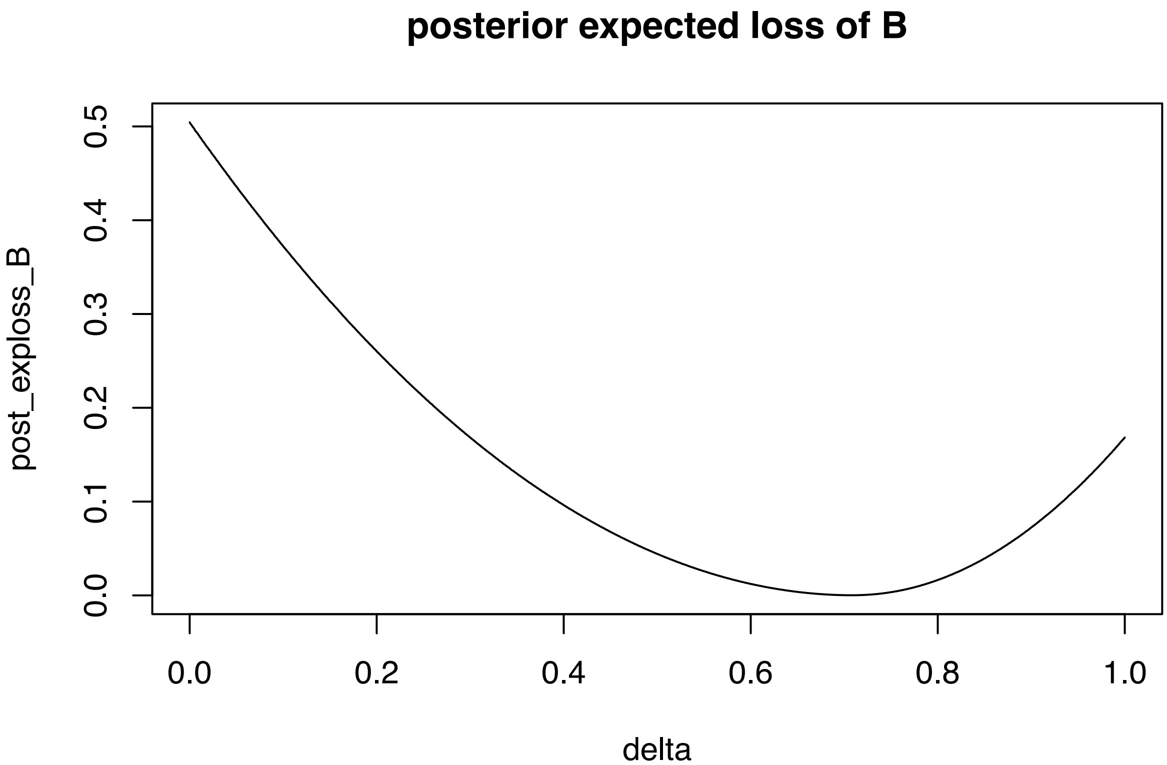


(d)

```
set.seed(1)
a.A = 712; b.A = 291; a.B = 713; b.B = 291
n = 1000; x = 710

loss_func = function(theta, delta){
  if (delta < theta){
    return((theta - delta)^2)} else{
    return(2 * (theta - delta)^2)
    }
  }

posterior_exploss = function(delta, S = 10000){
  theta = rbeta(S, a.A + x, b.A + n - x)
  loss = apply(as.matrix(theta), 1, loss_func, delta)
  risk = mean(loss)
}

delta = seq(0, 1, length = 1000)
post_exploss_A = apply(as.matrix(delta), 1, posterior_exploss)
plot(delta, post_exploss_A, type = "l", main ="posterior expected loss of A")
```




```
posterior_exploss = function(delta, S = 10000, a, b){
  theta = rbeta(S, a.B + x, b.B + n - x)
  loss = apply(as.matrix(theta), 1, loss_func, delta)
  risk = mean(loss)
}

post_exploss_B = apply(as.matrix(delta), 1, posterior_exploss)
plot(delta, post_exploss_B, type = "l", main ="posterior expected loss of B")
```




```
round(delta[which.min(post_exploss_A)], 2)
```

```
## [1] 0.71
```

```
round(delta[which.min(post_exploss_B)], 2)
```

```
## [1] 0.71
```

The bayes estimates for the two statisticians are the same.