Stats 102C, Homework 3 - Intro to MCMC

Jang, Junhyuk (004 728 134)

Review of Monte Carlo Integration

In the last homework, we estimated the integral of $h(x)$ by using Monte Carlo estimation.

$$h(x) = \exp(-0.5(x-2)^2 - 0.1|\sin(2x)|)$$

$$I = \int_0^5 h(x)dx = \int_0^5 \exp(-0.5(x-2)^2 - 0.1|\sin(2x)|)dx$$

We want to estimate the value of I. We say that the average value of $h(x)$ over the integral is:

$$\frac{1}{5}I = E_f[h(X)]$$

Recall that for MC integration, we estimate

$$E_f[h(X)] = \int_{\mathcal{X}} h(x)f(x)dx$$

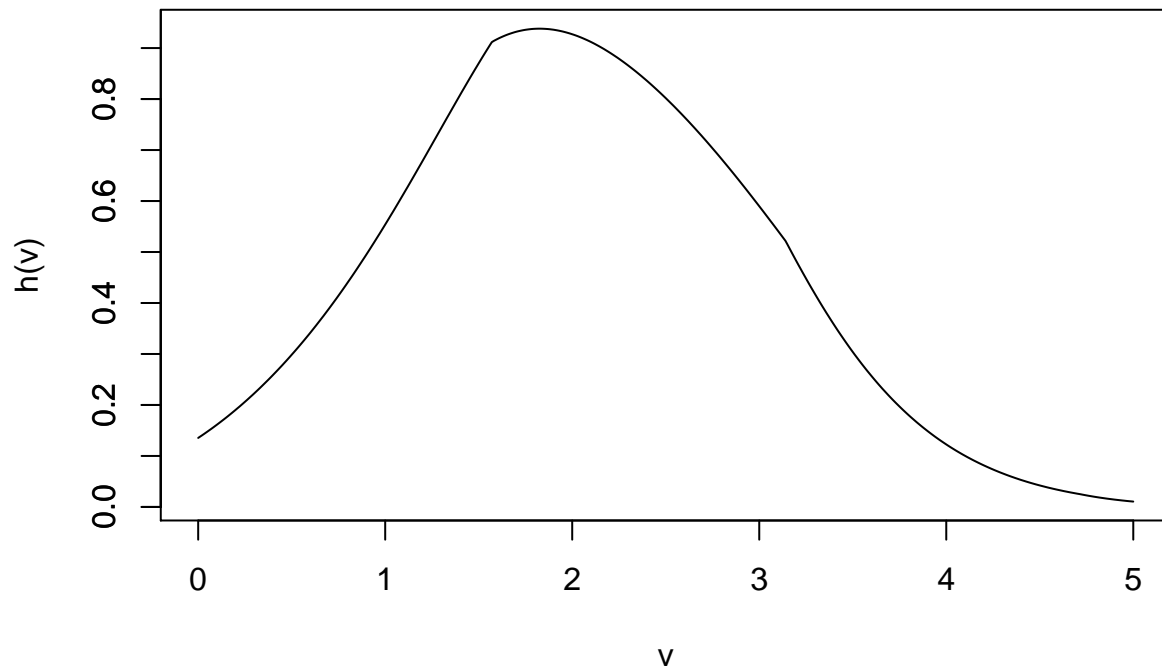Because we are using uniform sampling over the interval (0,5)

$$f(x) = 1/5$$

Thus,

$$\frac{1}{5}I = \int_0^5 h(x)f(x)dx \approx \frac{1}{N}\sum_{j=1}^N h(x_j)$$

Where $x_j \sim \text{Unif}(0,5)$

```
n = 5000
h <- function(x) exp(-0.5 * (x - 2) ^ 2 - 0.1 * abs( sin(2*x) ) )
v <- seq(0,5, by = 0.01)
plot(v, h(v), type = "l")
axis(side = 2, at = seq(0,1, by = 0.1))
```

When we sample randomly from the uniform distribution, every value between 0 and 5 have an equal chance of being selected.

Let's say I drew 15 x-values that were equally spaced from 0.1 to 4.9. To estimate the expected value of our function, I would calculate the value of the function at each x (shown in red below). I'd then take those values and find the mean of them. (Shown as green dots on the y-axis. I take the mean of those 15 green points.)

```r
x <- seq(0.1, 5, by = 0.3)
n <- length(x)
h_x <- h(x)

plot(v, h(v), type = "l", ylim = c(0,1))
axis(side = 2, at = seq(0,1, by = 0.1))
points(x, h_x, pch = 19, cex = 0.5, col = 'red')
points(rep(0,n), h_x, pch = 19, cex = 0.5, col = 'green')
segments(x, h_x, rep(0, n), h_x, lty = "dotted")
```
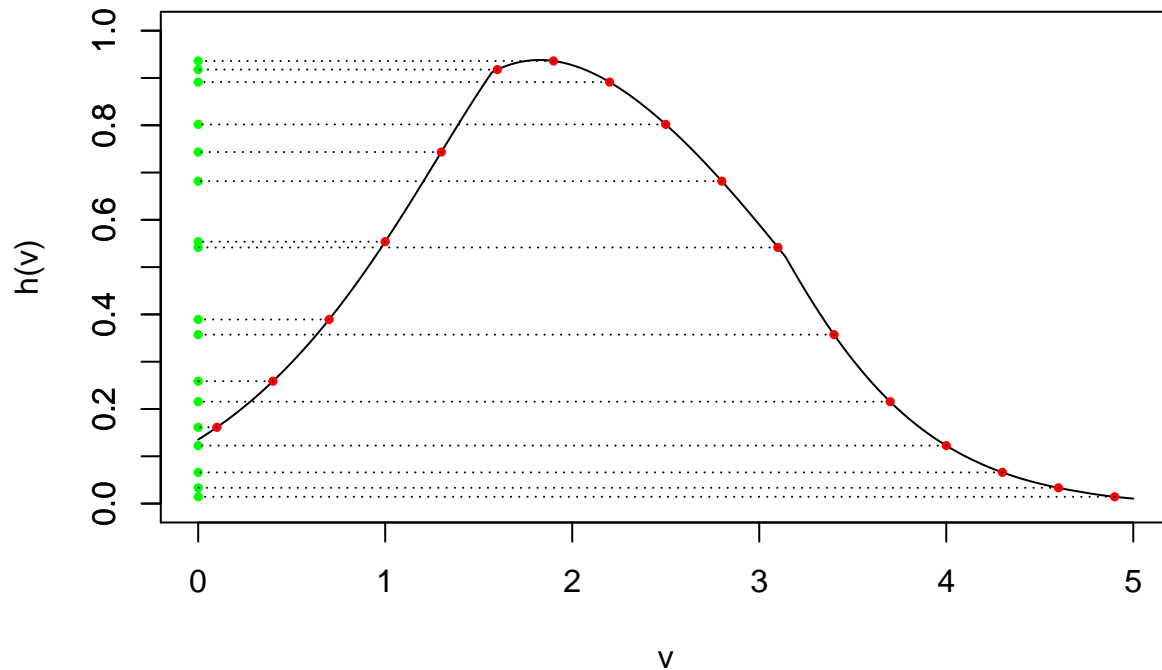
In this case, the values of those 15 green points are:

```
round(h_x, 4) # rounded to 4 decimal places for clarity
```

```
##  [1] 0.1612 0.2588 0.3892 0.5538 0.7434 0.9177 0.9360 0.8912 0.8018 0.6817
## [11] 0.5416 0.3572 0.2155 0.1226 0.0660 0.0333 0.0144
```

And the mean of these values is:

```
mean(h_x)
```

```
## [1] 0.4520836
```

And the estimate of the integral would be:

```
mean(h_x) * 5
```

```
## [1] 2.260418
```

If I draw **RANDOM** uniform samples, we can see how the estimate of μ fluctuates quite a bit before 'settling' down.

```
## Monte Carlo Integration using uniform random sampling
n = 200

# First series
set.seed(1)
x <- runif(n, 0, 5)
hbar_n <- cumsum(h(x))/c(1:n)
mu_n <- hbar_n * 5


plot(1:n, mu_n, type = "l", ylim = c(1, 4.5))
abline(h = 2.29583, lty='dotted', col = 'red')

# Second series using a different random seed
set.seed(2)
```
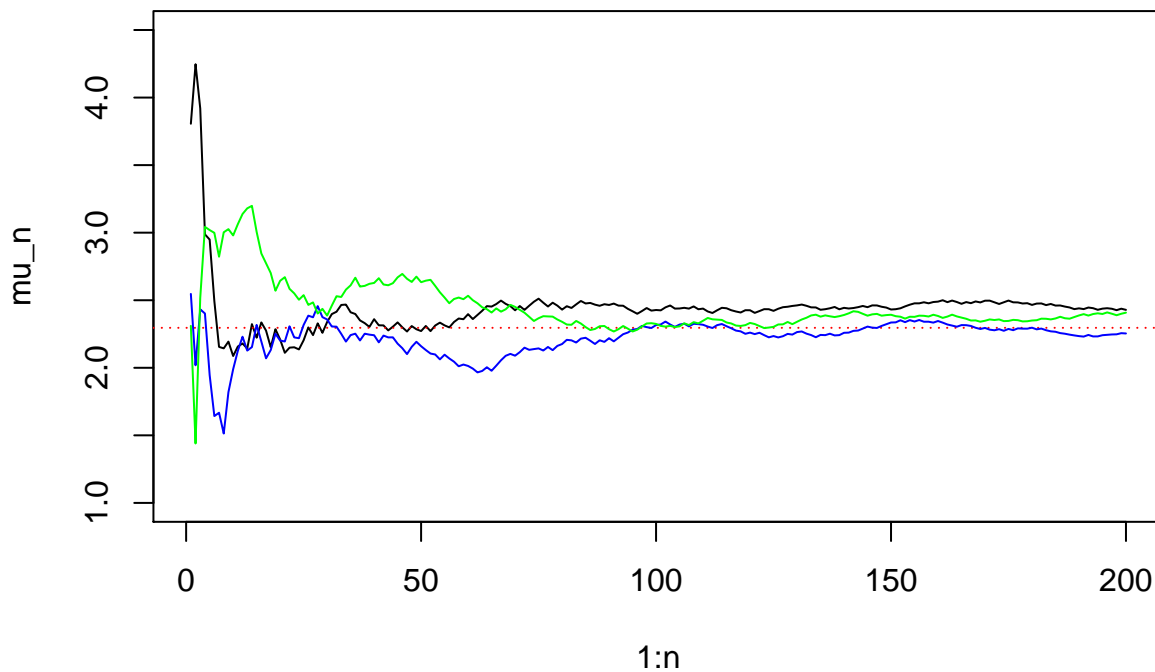
```r
x <- runif(n, 0, 5)
hbar_n <- cumsum(h(x))/c(1:n)
mu_n <- hbar_n * 5
lines(1:n, mu_n, type = "l", col = 'blue')

# Third series with different random seed point
set.seed(3)
x <- runif(n, 0, 5)
hbar_n <- cumsum(h(x))/c(1:n)
mu_n <- hbar_n * 5
lines(1:n, mu_n, type = "l", col = 'green')
```


The estimate of the integral $\hat{I}$ is .

```r
print(mu_n[n])
```

```
## [1] 2.408088
```

Importance Sampling (weighted monte carlo estimation)

With importance sampling, we don't draw from the distribution $f(x)$, but a trial distribution $g(x)$.

Even though it is easy to draw from $f(x)$, which is the uniform distribution, we may see that it can be advantageous to draw from a different distribution that more closely resembles the target function.

Importance Sampling

We will use the normal distribution N(2, 1) as the trial distribution g(x) to estimate the same integral by importance sampling.

$$I = \int_0^5 \exp[-0.5(x-2)^2 - 0.1|\sin(2x)|]dx$$

4

$$\frac{1}{5}I = E_f[h(X)] = \int_0^5 h(x)f(x)dx = \int_0^5 h(x)\frac{f(x)}{g(x)}g(x)dx \approx \frac{1}{N}\sum_{j=1}^{N} h(x_j)\frac{f(x_j)}{g(x_j)}$$

```r
# what the function looks like, along with the normal distribution
h <- function(x) exp(-0.5 * (x - 2) ^ 2 - 0.1 * abs( sin(2*x) ) )
v <- seq(0,5, by = 0.01)
norm_pdf <- dnorm(v, mean = 2, sd = 1)

# If we were doing accept-reject sampling, this is a way to find the optimal ratio to make
# sure our proposal distribution is always greater than our target distribution
optimize(f = function(x){  dnorm(x,2,1) / h(x) }, interval=c(0,5))
```

```
## $minimum
## [1] 3.141592
##
## $objective
## [1] 0.3989423
```

```r
# the function inside optimize is the proposal divided by the target.
# Optimize gives the location where the ratio between the proposal and target is smallest.
# If we multiply the proposal distribution by 1/ratio, then the proposal distribution will always be =
# distribution at this location. So we use this value as our constant M.
m <- optimize(f = function(x){  dnorm(x,2,1) / h(x) }, interval=c(0,5))$objective
# Technically, this does not matter at all for importance sampling, but it makes it easier to see that
# the trial distribution matches the desired function quite well.

plot(v, norm_pdf * (1/m), type = "l", col = "blue")  # trial distribution
lines(v, h(v), type = "l", col = "black")  # desired function
```
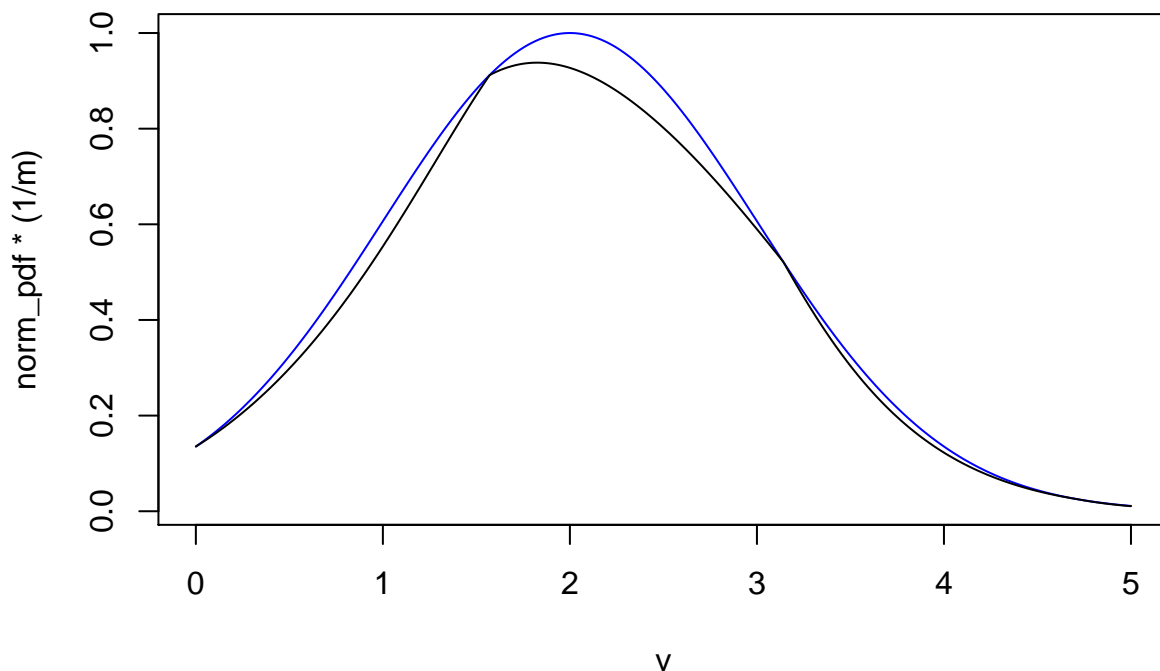


```r
# we see a pretty good match between the trial distribution and desired function.
```

We will use importance sampling to estimate $\hat{I}$. We will use `rnorm()` to generate random normal values.

Keep in mind that $f(x) = 1/5$.

$g(x)$ is the normal density with mean 2 and sd 1. However, we must throw away any values outside of the range (0, 5).

If we do that, then $g(x)$ is no longer a probability density because it will no longer integrate to 1.

$$\int_0^5 g(x) \neq \int_{-\infty}^{\infty} g(x) = 1$$

To fix this, we need to find a constant to multiply with $g(x)$ so that

$$\int_0^5 C \cdot g(x) = 1$$

Problem 1

Find C so that $\int_0^5 C \cdot g(x) = 1$. Let $g(x)$ be the normal density with mean 2 and sd 1. Hint: figure out how much of the distribution is 'cut off' at 0 and 5, and find C accordingly.

```
a <- pnorm(5,2,1) - pnorm(0,2,1)
(C <- 1/a)
```

```
## [1] 1.024695
```

Problem 2

Use the code in the code chunk 'threeMCchains' as a starting point.

Change runif to rnorm. Make sure you remove values of x below 0 and above 5. Adjust how $\bar{h}_n$ is calculated according to importance sampling. Then estimate the integral.

When you create your plot, also adjust the axes to fit the samples better.

Finally, comment on how quickly the method using importance sampling converges to the expected value versus the uniform sampling method.

```
## Monte Carlo Integration using importance sampling
n = 200

# First series
set.seed(1)
plot.new()
x <- rnorm(1000,2,1)
length(x[(x >= 0) & (x <= 5)])
```

```
## [1] 965
```

```
fil_x <- x[(x >= 0) & (x <= 5)][1:200]
hbar_n <- cumsum(h(fil_x) / (C*dnorm(fil_x,2,1)) * (1/5)) / c(1:n)
mu_n <- hbar_n * 5

plot(1:n, mu_n, type = "l", ylim = c(2.2, 2.4))
abline(h = 2.29583, lty='dotted', col = 'red')


# Second series using a different random seed
set.seed(2)
```

```r
x <- rnorm(1000,2,1)
length(x[(x >= 0) & (x <= 5)])
```
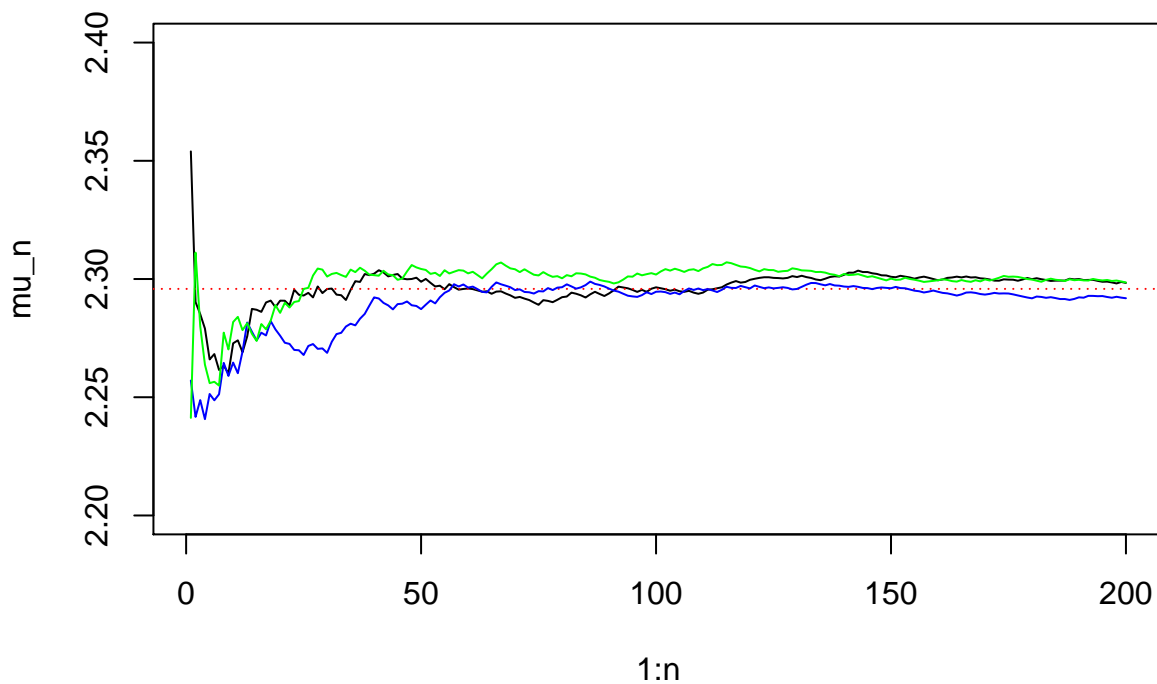
[1] 979

```r
fil_x <- x[(x >= 0) & (x <= 5)][1:200]
hbar_n <- cumsum(h(fil_x) / (C*dnorm(fil_x,2,1)) * (1/5)) / c(1:n)
mu_n <- hbar_n * 5
lines(1:n, mu_n, type = "l", col = 'blue')

# Third series with different random seed point
set.seed(3)
x <- rnorm(1000,2,1)
length(x[(x >= 0) & (x <= 5)])
```

[1] 976

```r
fil_x <- x[(x >= 0) & (x <= 5)][1:200]
hbar_n <- cumsum(h(fil_x) / (C*dnorm(fil_x,2,1)) * (1/5)) / c(1:n)
mu_n <- hbar_n * 5
lines(1:n, mu_n, type = "l", col = 'green')
```



```r
# we can easily see that method using importance sampling converges
# to the expected value much more faster than the uniform sampling method.
```

Bayesian Thinking

Problem 3: Doing Bayesian Data Analysis: Exercise 5.1

```r
# person has disease is denoted by D = +
# person does not has disease is denoted by D = -
# Test positive is denoted by T = +
# Test negative is denoted by T = -
```

```
# first test
# p(D = + | T = +)
# = p(T=+|D=+) * p(D=+) / (p(T=+|D=+) * p(D=+) + p(T=+|D=-) * p(D=-))
(poster <- (0.99 * 0.001) / ((0.99 * 0.001) + (0.05 * (1 - 0.001))))
```

```
## [1] 0.01943463
```

```
# second test
prior <- poster
# p(D = + | T = -) when first test was positive
# = p(T=-|D=+) * prior / (p(T=-|D=+)*prior + p(T=-|D=-) * (1-prior))
(prob <- (0.01 * prior) / ((0.01 * prior) + 0.95 * (1-prior)))
```

```
## [1] 0.0002085862
```

Problem 4: Doing Bayesian Data Analysis: Exercise 5.3

```
# first test
# p(D = + | T = -)
# = p(T=-|D=+) * p(D=+) / (p(T=-|D=+) * p(D=+) + p(T=-|D=-) * p(D=-))
(poster <- (0.01 * 0.001) / ((0.01 * 0.001) + (0.95 * (1 - 0.001))))
```

```
## [1] 1.053674e-05
```

```
# second test
prior <- poster
# p(D = + | T = +) when first test was negative
# = p(T=+|D=+) * prior / (p(T=+|D=+)*prior + p(T=+|D=-) * (1-prior))
(prob <- (0.99 * prior) / ((0.99 * prior) + (0.05 * (1-prior))))
```

```
## [1] 0.0002085862
```

```
# We can notice that probability of first test result was postive and second
# test result was negative for a person who has disease is same as
# the probability of first test result was negative and second test result was
# positive for a person who has disease.
```

Modeling the Beta-Binomial Model with grid approximation

In Bayesian inference, we often write the posterior distribution of some parameter θ based on the data y as follows:

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

We label $P(y|\theta)$ the *likelihood* of the data given the value of the parameter θ.

$P(\theta)$ represents our *prior* distribution of the possible parameter values of θ.

$P(y)$ is the *marginal* distribution of the observed data y. This is generally found by summing or integrating the joint probability of the data y and parameter θ across all possible values of θ. In many cases, this integral is intractable. The good news is that it is just a constant.

As such, we often say that the posterior distribution is proportional to the numerator.

$$P(\theta|y) \propto P(y|\theta)P(\theta)$$

Summary of Ch 6

If the beta distribution prior has distribution $\text{Beta}(\alpha, \beta)$

And our data has z successes, and $N - z$ failures, the posterior distribution will have distribution:

$$\text{Beta}(z + \alpha, N - z + \beta)$$

Let's further explore the relationship between the prior, the likelihood, and the posterior distributions.

The beta prior for baseball batting average

Read: http://varianceexplained.org/statistics/beta_distribution_and_baseball/

As seen in the blog article, we will model the prior distribution of baseball batters' batting average as $\text{Beta}(81, 219)$

To emphasize that we are doing grid approximation, I am plotting the distribution as points

```
s <- seq(0.0, 1, by = 0.005)
plot(s, dbeta(s, 81, 219), pch = 19, cex = 0.2, ylab = 'density')
arrows(qbeta(0.025, 81, 219), 0.5, qbeta(0.975, 81, 219), 0.5, col = 'red', code = 3, angle = 90, length
```
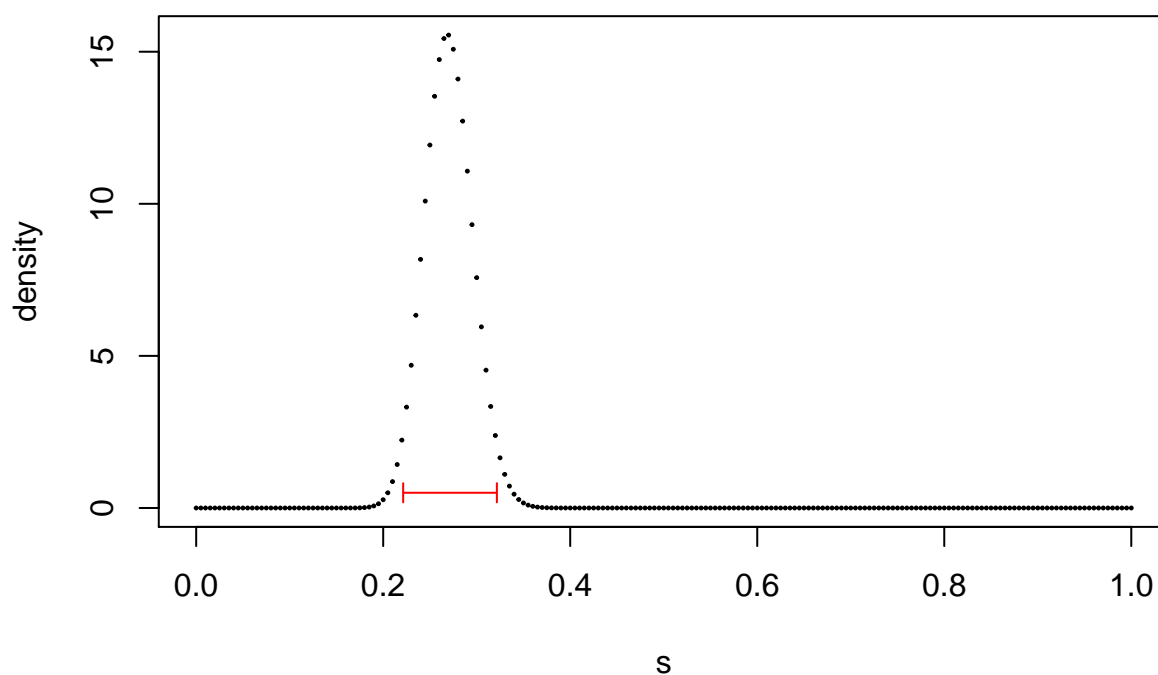

Credibility interval:

```
print( c( qbeta(0.025, 81, 219), qbeta(0.975, 81, 219) ) )  # equal tailed Credibility interval
```

```
## [1] 0.2213490 0.3215554
```

Before seeing any data, my prior distribution tells me that there is a 95% probability that the batter's batting average is between 0.2213 and 0.3216.
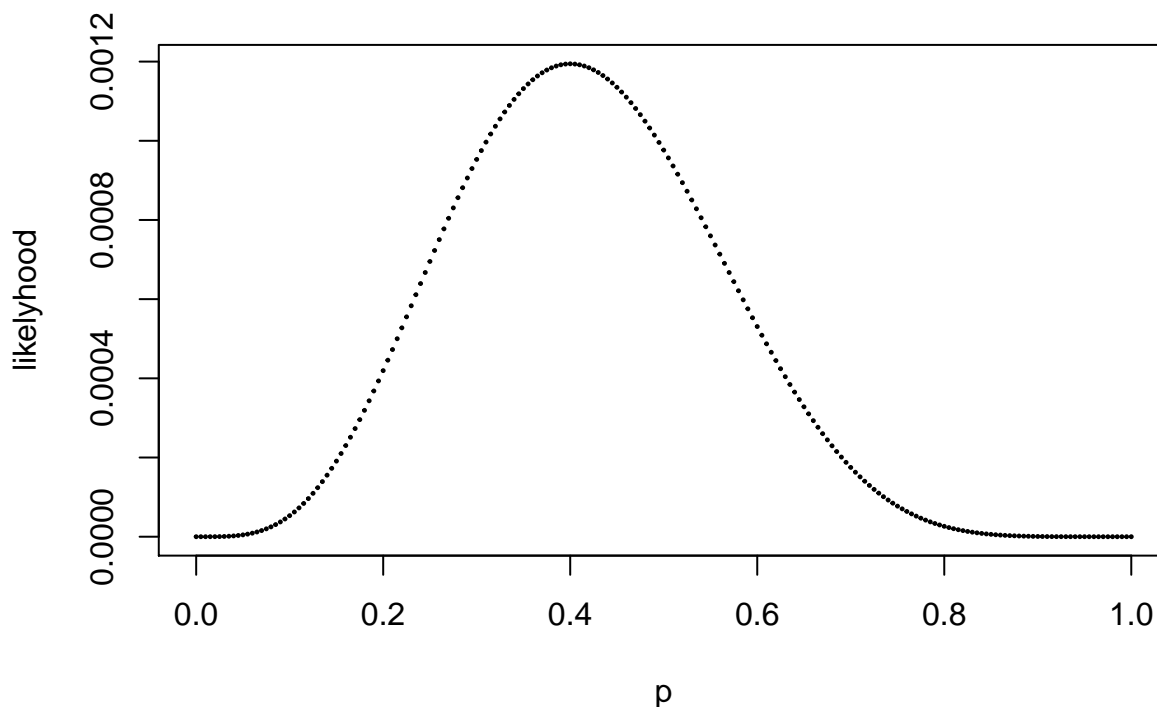
Problem 5

Let's say you observe a player who had 10 at bats and has 4 base hits.

Plot the likelihood of the data for values of p between 0.0 and 1. Use the same vector `s` for the locations of the grid approximation.

```
posterior <- function(theta) {
  (theta)^4 * (1-theta)^6
}

plot(s,posterior(s),pch = 19, cex = 0.2, ylab ='likelyhood',xlab='p')
```


Use the known results for the posterior distribution: $\text{Beta}(z + \alpha, N - z + \beta)$. Plot the posterior distribution of p after considering the data (use points, rather than a line to emphasize that we are using grid approximation). Use red points for the posterior. Also plot the prior distribution in black. You will see just a slight shift between the prior and the posterior.

```
plot(s,dbeta(s,81,219),pch = 19, cex = 0.2, ylab = 'density',xlab = 'p')
points(s,dbeta(s,85,225),pch = 19, cex = 0.2, ylab = 'density',col = 'red')
points(s,posterior(s),pch = 19, cex = 0.2,col = 'green')
```
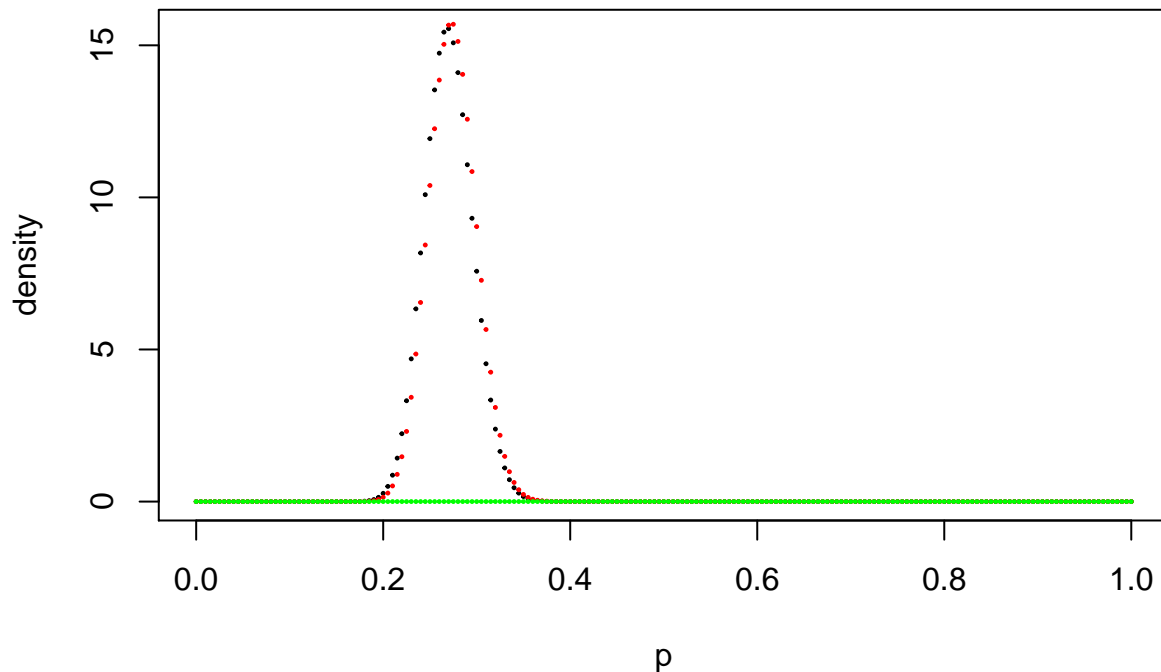
Use `qbeta()` to create a 95% credibility interval based on the posterior distribution.

```
(cred_interval <- c( qbeta(0.025, 85, 225), qbeta(0.975, 85, 225) ) )
```

```
## [1] 0.2260414 0.3251052
```

Use classical statistics to create a 95% confidence interval for p based on the fact that you had 4 successful hits out of 10 trials. (Even though the large sample condition is not met, assume you can use the central limit theorem for the creation of the confidence interval.)

```
p_hat <- 4/10
bound <- sqrt((p_hat * (1-p_hat))/10)
upper <- p_hat + qnorm(0.975) * bound
lower <- p_hat - qnorm(0.975) * bound
(confidence_interval <- c(lower,upper))
```

```
## [1] 0.09636369 0.70363631
```

Add both the credibility interval (in red at the level y = 0.5) and the confidence interval (in blue at the level y = 0.6) to the plot so you can make a visual comparison.

```
plot(s,dbeta(s,81,219),cex = 0.2, ylab = 'density',xlab = 'p')
points(s,dbeta(s,85,225),pch = 19, cex = 0.2, ylab = 'density',col = 'red')
arrows(cred_interval[1],0.5,cred_interval[2],0.5,col = 'red',
       code = 3,angle = 90,length = 0.05)
arrows(confidence_interval[1],0.6,confidence_interval[2],0.6,col = 'blue',
       code = 3,angle = 90,length = 0.05)
```
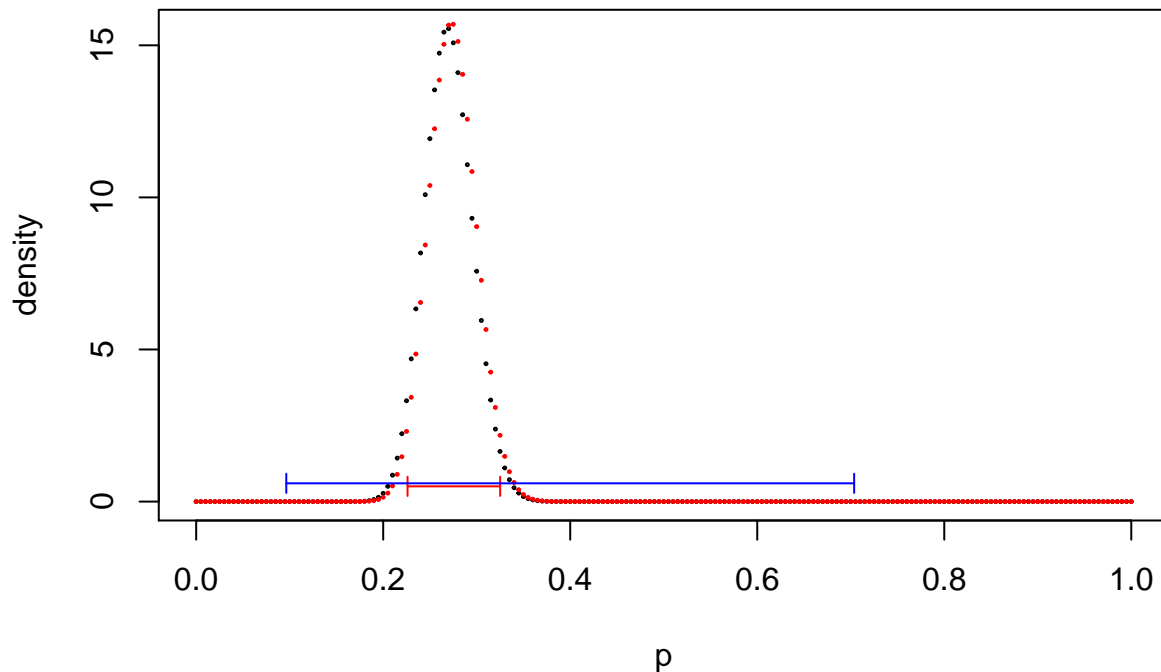
Problem 6a

Let's say you observe a player who had 100 at bats and has 35 base hits.

Plot the posterior distribution of p after considering the data (in red). Also plot the prior (in black). Comment on the difference between the prior and the posterior.

Find a 95% credibility interval based on the posterior. Create a classical 95% confidence interval. Compare the two intervals.

Add both the credibility interval (in red, at y = 0.5) and the confidence interval (in blue, at y = 0.6) to the plot so you can make a visual comparison.

```
p_hat <- 35/100
lower <- p_hat - 1.96 * sqrt(0.35*0.65/100)
upper <- p_hat + 1.96 * sqrt(0.35*0.65/100)

(cred_interval <- c( qbeta(0.025, 81+35, 219+65), qbeta(0.975, 81+35, 219+65) ) )
```
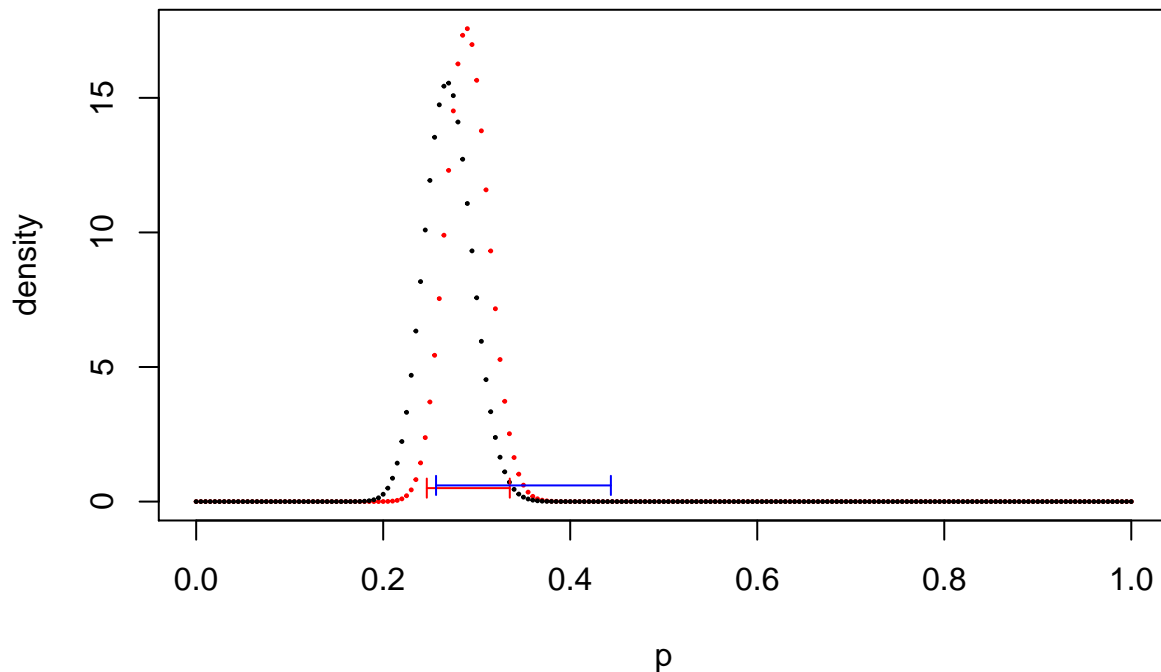
```
## [1] 0.2466139 0.3353751
```

```
(confide_interval <- c(lower,upper))
```

```
## [1] 0.256514 0.443486
```

```
plot(s,dbeta(s,81+35,219+65),pch = 19, cex = 0.2, ylab = 'density',col = 'red',
     xlab = 'p')
points(s,dbeta(s,81,219),pch = 19, cex = 0.2, ylab = 'density',col = 'black')
arrows(qbeta(0.025,81+35,219+65),0.5,qbeta(0.975,81+35,219+65),0.5,col = 'red',
code = 3,angle = 90,length = 0.05)
arrows(lower,0.6,upper,0.6,col = 'blue',code = 3, angle = 90,length = 0.05)
```

```
# We can see that the posterior distribution shifted to the right compared to the
# prior distriubution which implies that posterior has higer p on average.
# Also we can see that only half of confidence interval range is within the
# credibility interval. It is strongly expected to find p where confidence
# interval intersects with the credibility interval.
```

Problem 6b

Let's say you observe a player who had 500 at bats and has 175 base hits.

Plot the posterior distribution of p after considering the data. Also plot the prior. Comment on the difference between the prior and the posterior.

Find a 95% credibility interval based on the posterior. Create a classical 95% confidence interval. Compare the two intervals.

Add both the credibility interval (in red, at y = 0.5) and the confidence interval (in blue, at y = 0.8) to the plot so you can make a visual comparison.

```r
p_hat <- 35/100
lower <- p_hat - 1.96 * sqrt(0.35*0.65/500)
upper <- p_hat + 1.96 * sqrt(0.35*0.65/500)

(cred_interval <- c( qbeta(0.025, 81+175, 219+325), qbeta(0.975, 81+175, 219+325) ) )
```
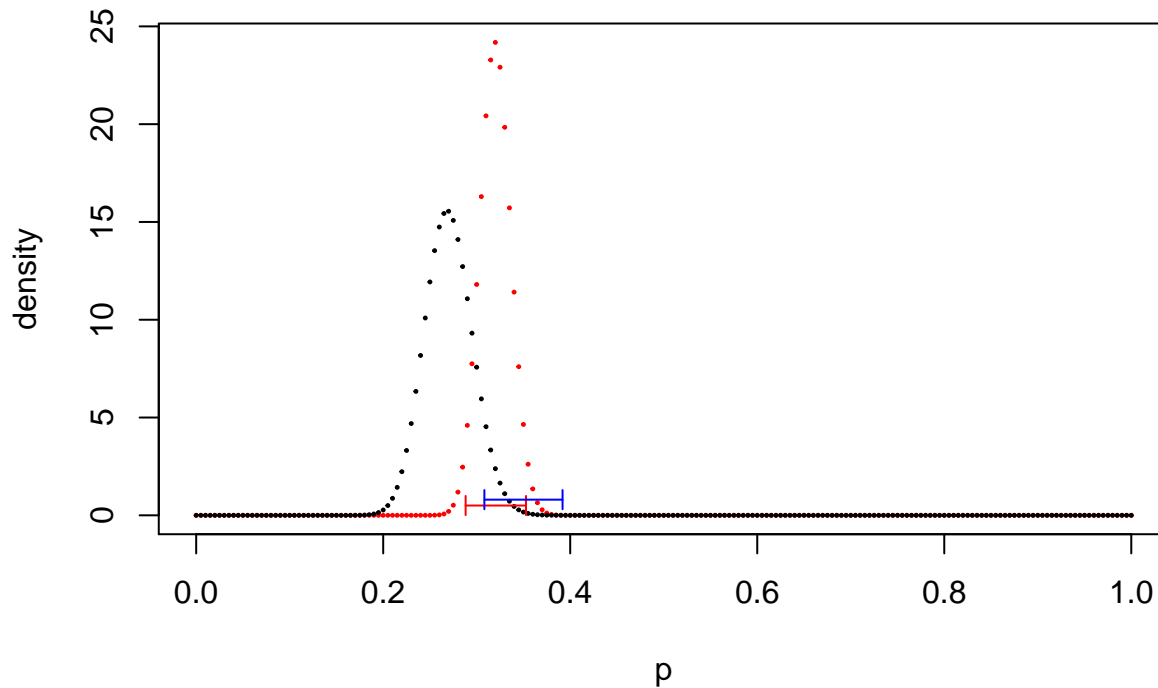
```
## [1] 0.2881325 0.3527200
```

```r
(confid_interval <- c(lower,upper))
```

```
## [1] 0.3081918 0.3918082
```

```r
plot(s,dbeta(s,81+175,219+325),pch = 19, cex = 0.2, ylab = 'density',col = 'red',
     xlab = 'p')
points(s,dbeta(s,81,219),pch = 19, cex = 0.2, ylab = 'density',col = 'black')
arrows(qbeta(0.025,81+175,219+325),0.5,qbeta(0.975,81+175,219+325),0.5,col = 'red',
```

```
code = 3,angle = 90,length = 0.05)
arrows(lower,0.8,upper,0.8,col = 'blue',code = 3, angle = 90,length = 0.05)
```



```
# we can see that the posterior distribution shifted more to the right and its
# amplitude has increased. Also, increased n makes confidence interval small.
# It is strongly expected to find p where the confidence interval intersects with
# the credibility interval.
```

Problem 6c

Finally, let's say you observe a player who had 5000 at bats and has 1750 base hits.

Plot the posterior distribution of p after considering the data. Also plot the prior.

Add both the credibility interval (in red, at y = 0.5) and the confidence interval (in blue, at y = 1) to the plot so you can make a visual comparison.

```
p_hat <- 35/100
lower <- p_hat - 1.96 * sqrt(0.35*0.65/5000)
upper <- p_hat + 1.96 * sqrt(0.35*0.65/5000)

(confide_interval <- c(lower,upper))
```

```
## [1] 0.3367791 0.3632209
```

```
(cred_interval <- c(qbeta(0.025, 81+1750, 219+3250), qbeta(0.975, 81+1750, 219+3250)))
```

```
## [1] 0.3327267 0.3583272
```

```
plot(s,dbeta(s,81+1750,219+3250),pch = 19, cex = 0.2, ylab = 'density',col = 'red',
     xlab = 'p')
points(s,dbeta(s,81,219),pch = 19, cex = 0.2, ylab = 'density',col = 'black')
arrows(qbeta(0.025,81+1750,219+3250),0.5,qbeta(0.975,81+175,219+325),0.5,col = 'red',
code = 3,angle = 90,length = 0.05)
arrows(lower,1,upper,1,col = 'blue',code = 3, angle = 90,length = 0.05)
```
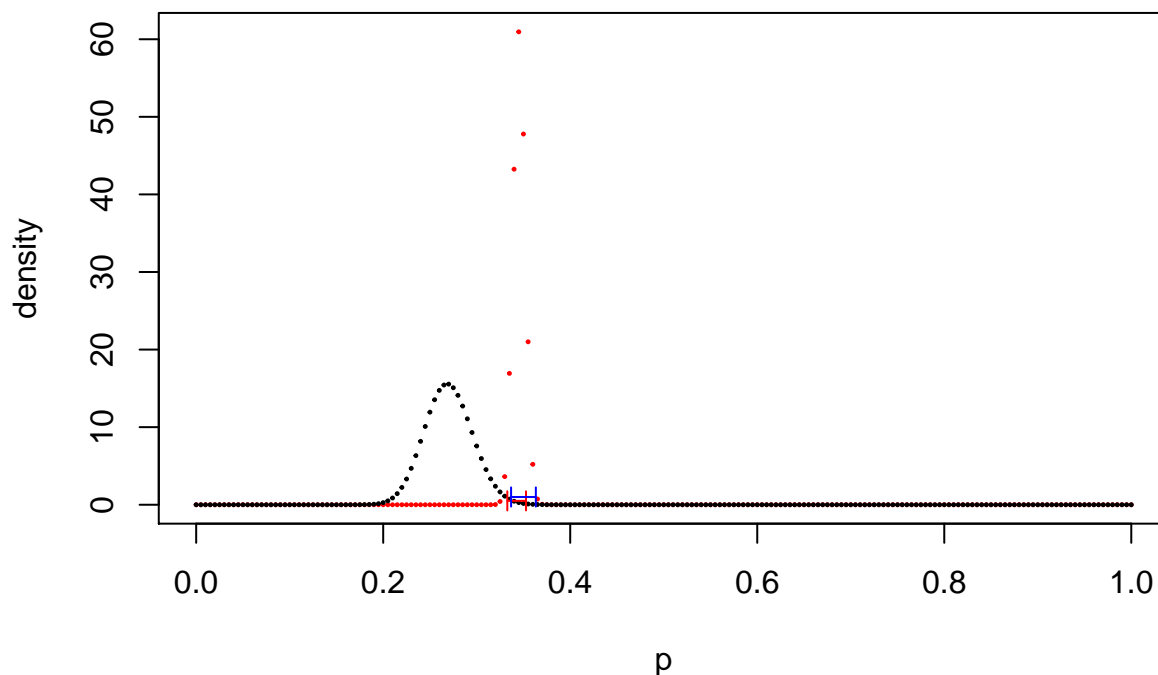
```
# we can see that large n makes confidence interval and credibility almost alway
# intersect. # It is strongly expected to find p where the confidence interval
# intersects with the credibility interval.
```

As the amount of data increases, how do the results of the Bayesian credibility interval compare to the results of the classical confidence interval?

```
# As the amount of data increases, the results of the Bayesian credibility interval
# start to match up with the results of the classical confidence interval.
```

Problem 7: Doing Bayesian Data Analysis: Exercise 6.5

When it says use the prior from exercise 6.4, they are refering to a Beta(0.1, 0.1) prior, as it appears in Figure 6.1.

When it asks, 'what is the predicted probability of heads for the 11th flip?', they are asking for the expected values (mean) of the posterior distribution of p.

```
# (A)
# Since our coin is minted by government, we have really strong
# prior belief that the coin is fair. So I will set my prior distribution as
# beta(1000,1000). The expected value of my prior distribution is 1000/2000 = 0.5.
# The posterior distribtion after 9 heads and 1 tails has expected value of
# 1009/2010 = 0.50199 which is not really different from the original expected value.

# (B)
# I will set my beta(0.1,0.1) since we have a strong prior belief that the coin is
# biased either strongly heads or tails. The expected value of my prior distribution
# is 0.1/0.2 = 0.5. The posterior distribtion after 9 heads and 1 tails has
# expected value of 9.1/10.2 = 0.8921569 which has high expected value of heads.
```