

Assignment

Ehsan Eslmai Shafigh

2023-05-07

Assignment 3

Exercise 1:

we start by initializing some common parameters:

```
n = 20
r = 7
n.sample = 1000
delta.p = 1/n.sample

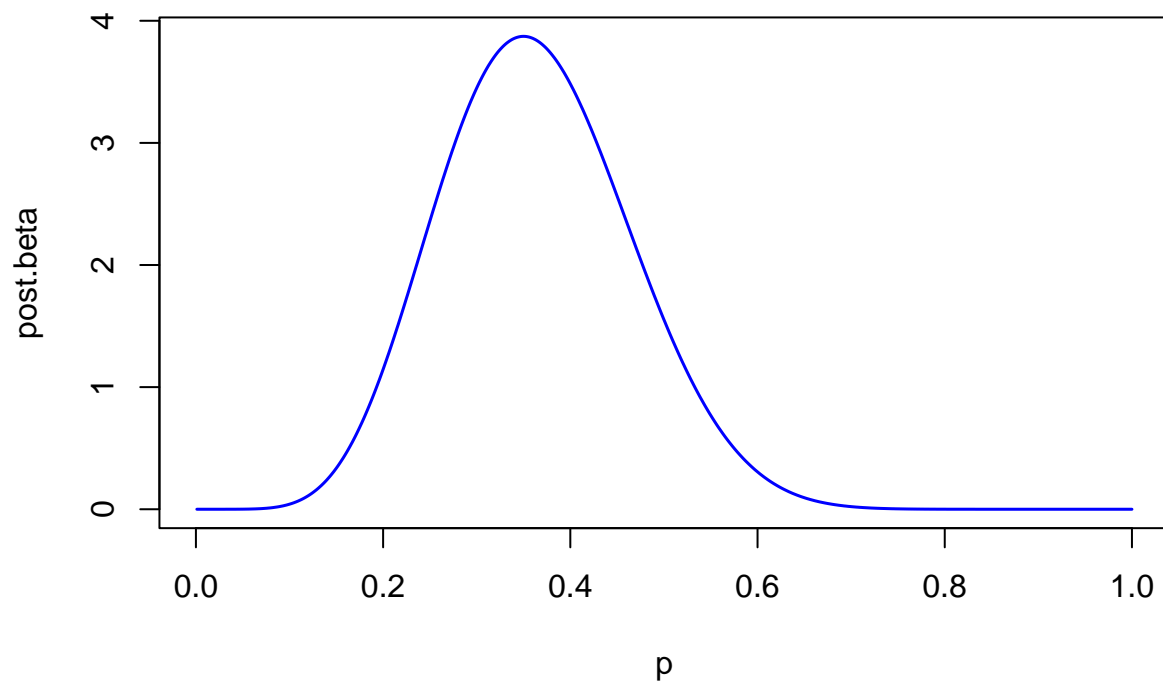
p = seq(from = 1/n.sample , by = delta.p , length.out = n.sample)
```

Now we start by assuming a uniform prior:

```
alpha.prior = 1
beta.prior = 1

post.beta = dbeta(x = p , alpha.prior + r , beta.prior + n - r )
```

```
plot(p , post.beta , type = 'l' , lwd = 1.5 , col = 'blue')
```



The summary of the results is as follows:

```
p.mom1 = delta.p * sum(p * post.beta)
p.mom2 = delta.p * sum(p^2 * post.beta)

print(p.mom1)
```

```
## [1] 0.3636364
```

```
print(p.mom2)
```

```
## [1] 0.1422925
```

The 95 credibility interval can be computed as follows:

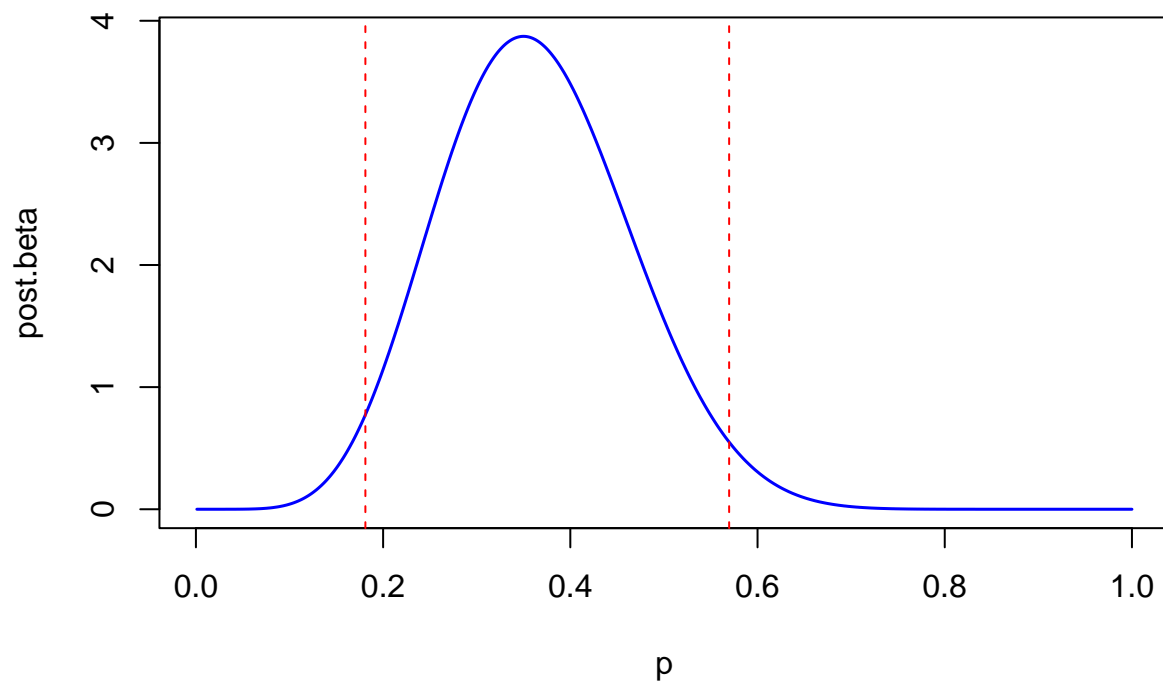
```
ci = qbeta(c(0.025 , 0.975), alpha.prior + r , beta.prior + n - r)

ci
```

```
## [1] 0.1810716 0.5696755
```

```
plot(p , post.beta , type = 'l' , lwd = 1.5 , col = 'blue')

abline(v = ci[1], lty = "dashed", col = "red")
abline(v = ci[2], lty = "dashed", col = "red")
```

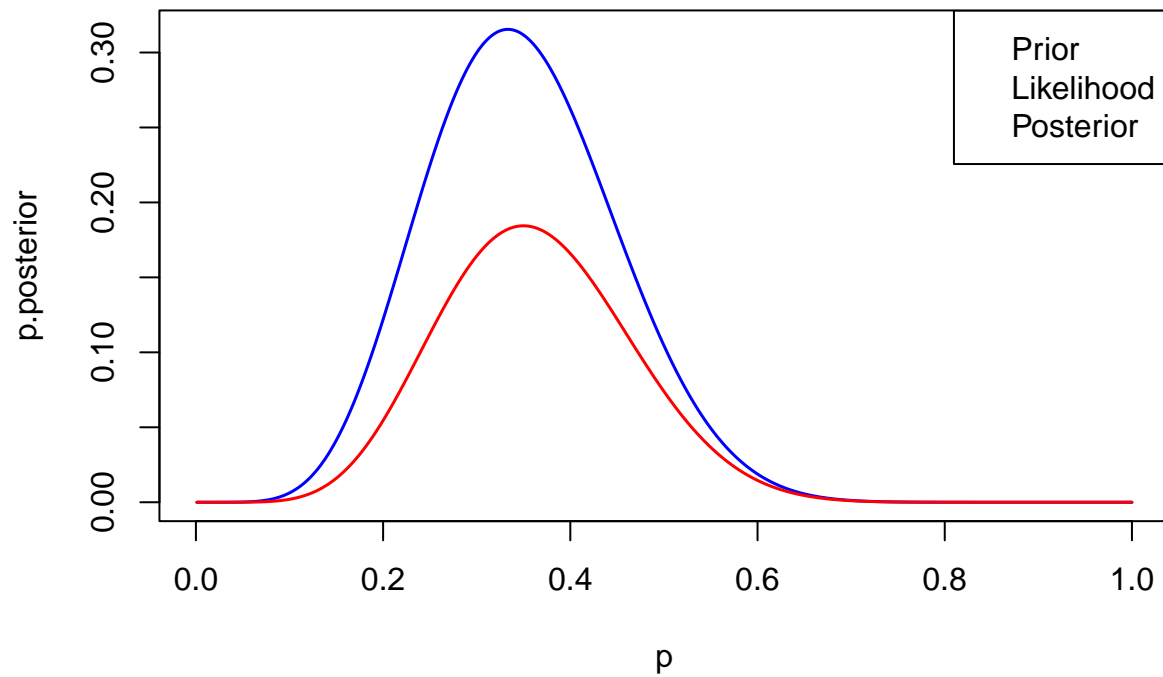


for Jeffery's prior we have:

```
p.prior = 1/sqrt(p)
likelihood = dbinom(x = r , size = n , prob = p)
p.posterior = p.prior * likelihood
```

```
plot(p , p.posterior , type = 'l' , lwd = 1.5 , col = 'blue')
lines(p , likelihood , type = 'l' , lwd = 1.5 , col = 'red' )
lines(p , p.prior , type = 'l' , lwd = 1.5 , col = 'black')
```

```
legend('topright' , legend = c('Prior', 'Likelihood', 'Posterior') , col = c('black', 'red', 'blue'))
```



```
p.posterior = p.posterior/sum(p.posterior)
```

```
p.mom1 = sum(p * p.posterior)
p.mom2 = sum(p^2 * p.posterior)
```

```
print(p.mom1)
```

```
## [1] 0.3488372
```

```
print(p.mom2)
```

```
## [1] 0.1317829
```

```
samples = sample(p , size = 1000 , replace = TRUE , prob = p.posterior)
```

```
ci = quantile(samples , c(0.025 , 0.975))
```

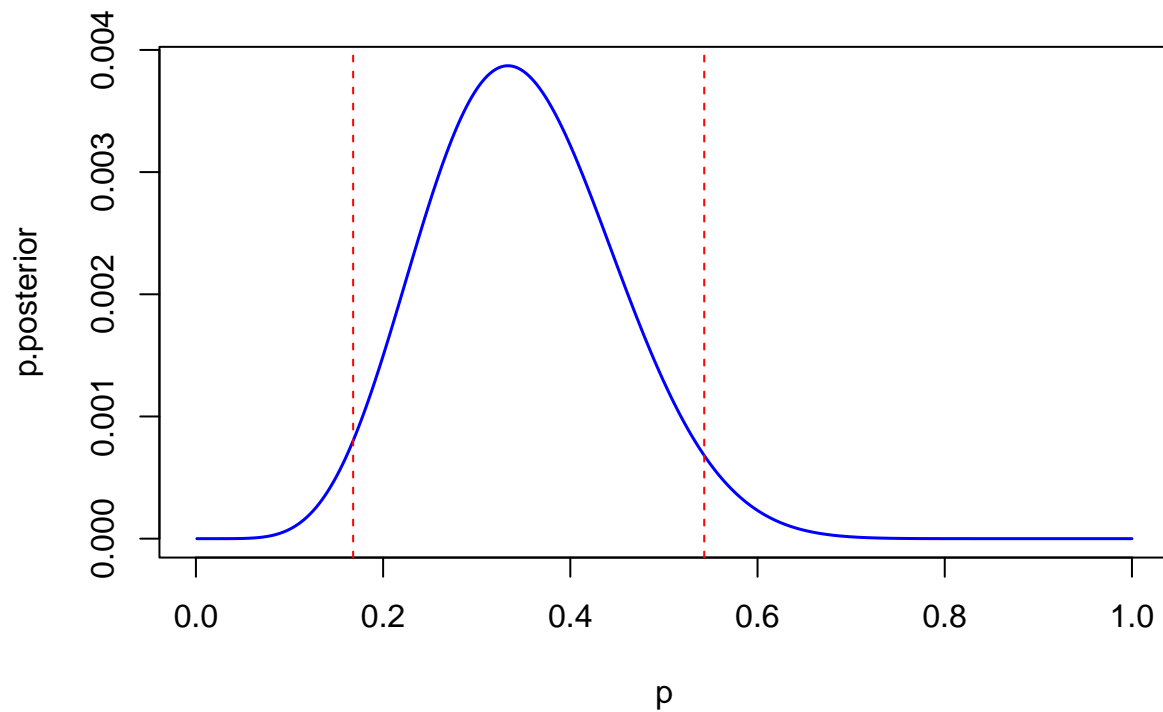
```
ci
```

```
##      2.5%      97.5%
```

```
## 0.167975 0.543025
```

```
plot(p , p.posterior , type = 'l' , lwd = 1.5 , col = 'blue')

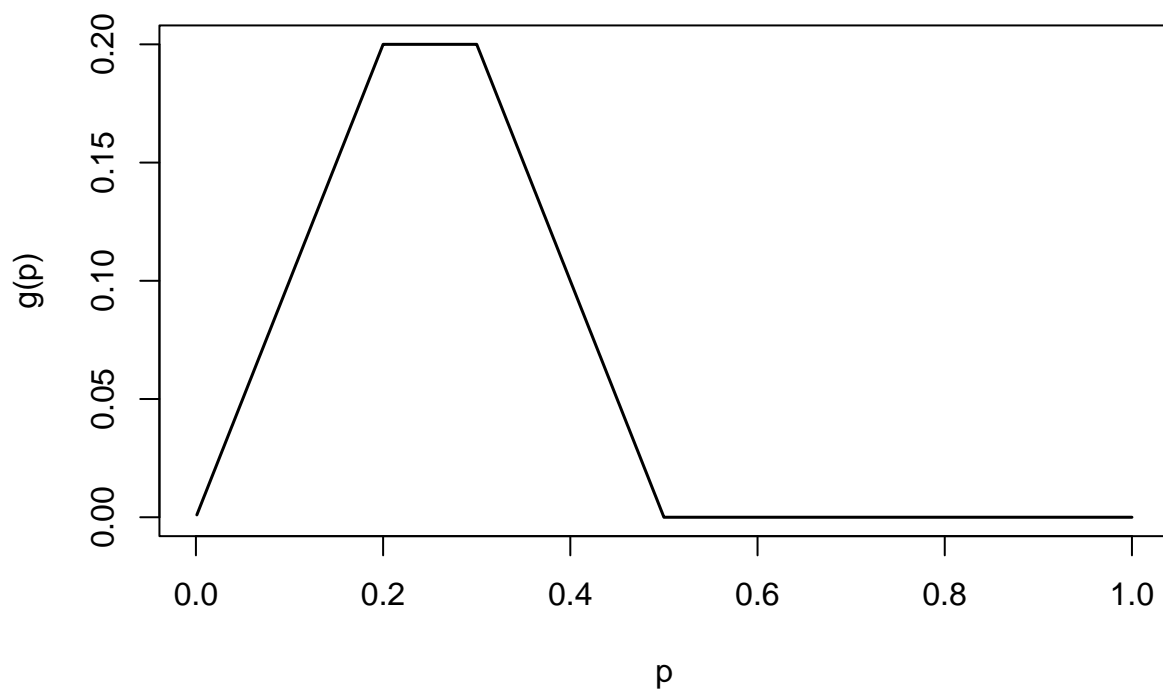
abline(v = ci[1], lty = "dashed", col = "red")
abline(v = ci[2], lty = "dashed", col = "red")
```



For a prior as the step function given, first we need to define the function:

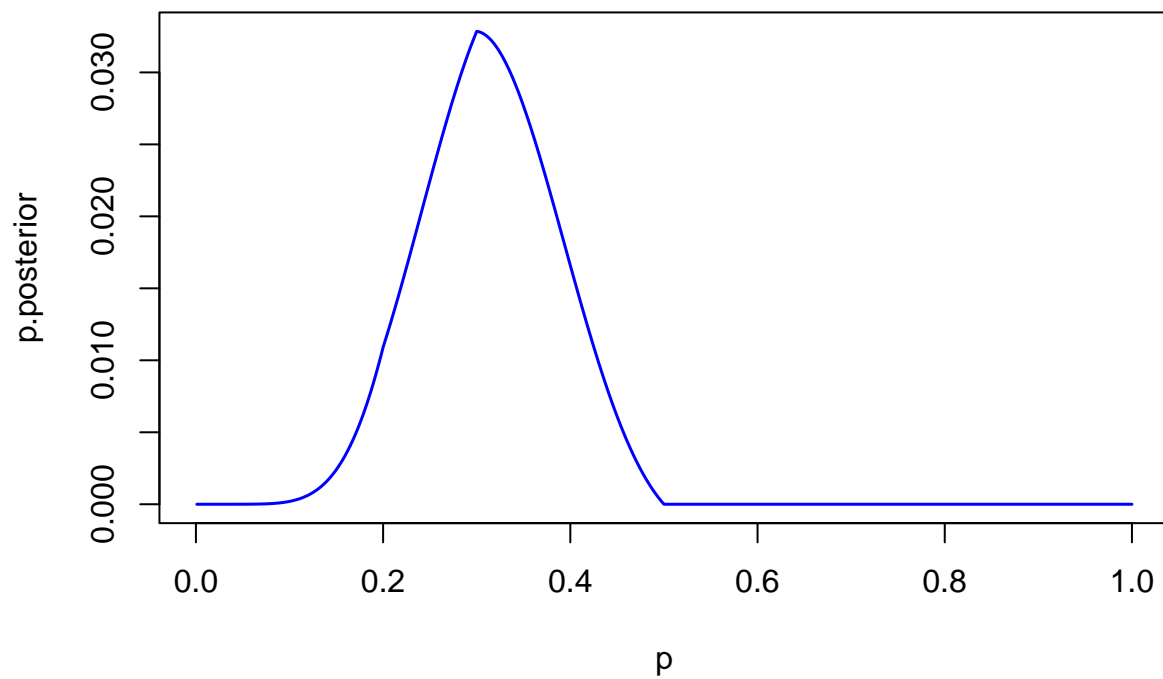
```
g = function(x){ifelse(x <= 0.2 , x , ifelse(x>0.2& x<=0.3 , 0.2 ,ifelse(x>0.3 & x<=0.5 , 0.5 - p , 0))}

plot(p , g(p) , type = 'l' , lwd = 1.5 , col = 'black')
```



```
p.prior = g(p)
likelihood = dbinom(x = r , size = n , prob = p)
p.posterior = p.prior * likelihood
```

```
plot(p , p.posterior , type = 'l' , lwd = 1.5 , col = 'blue')
```



```
p.posterior = p.posterior/sum(p.posterior)
```

```
p.mom1 = sum(p * p.posterior)
p.mom2 = sum(p^2 * p.posterior)
```

```
print(p.mom1)
```

```
## [1] 0.3120452
```

```
print(p.mom2)
```

```
## [1] 0.1023709
```

```
samples = sample(p , size = 1000 , replace = TRUE , prob = p.posterior)
```

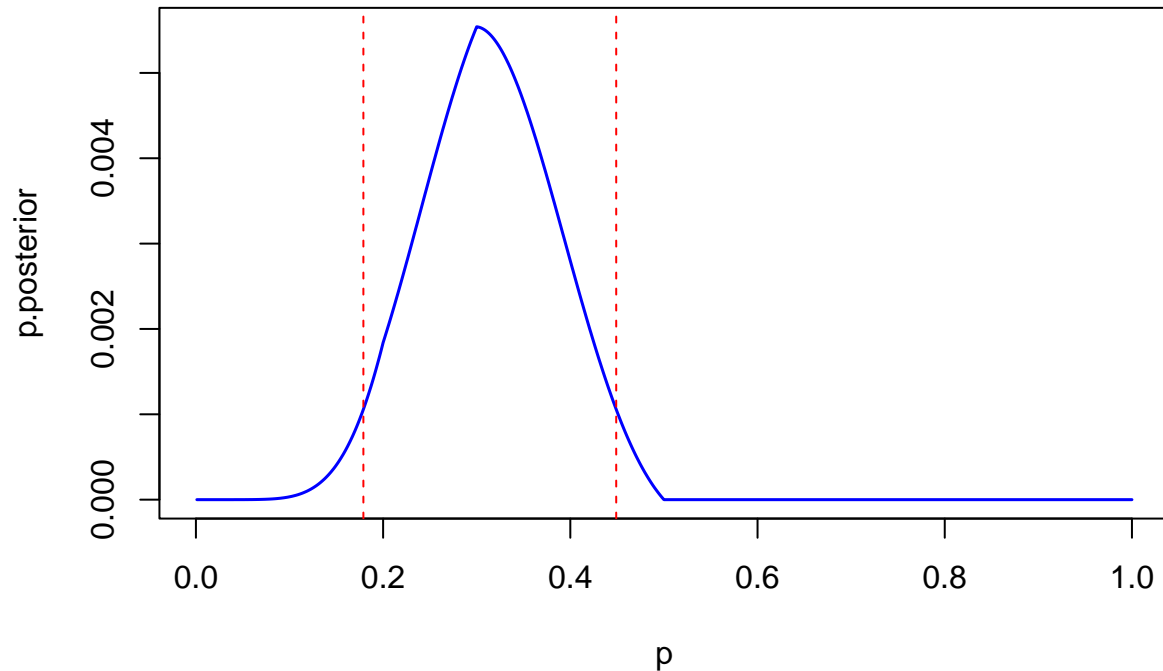
```
ci = quantile(samples , c(0.025 , 0.975))
```

```
ci
```

```
##      2.5%      97.5%
```

```
## 0.178925 0.449025
```

```
plot(p , p.posterior , type = 'l' , lwd = 1.5 , col = 'blue')
abline(v = ci[1], lty = "dashed", col = "red")
abline(v = ci[2], lty = "dashed", col = "red")
```



Exercise 2:

We can assume a binomial model for the posterior distribution of π .

```
n = 116
r = 17
n.sample = 1000
delta.p = 1/n.sample

p = seq(from = 0 , by = delta.p , length.out = n.sample)
```

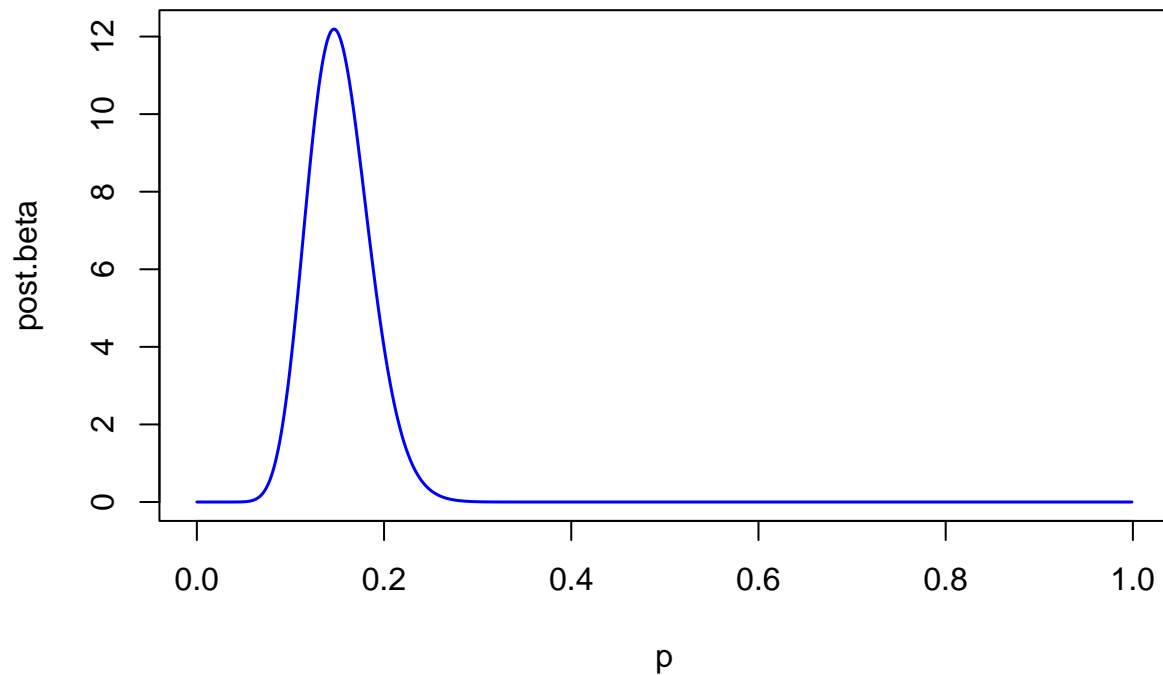
Starting with a uniform prior we have:

```
alpha.prior = 1
beta.prior = 1

post.beta = dbeta(x = p , alpha.prior + r , beta.prior + n - r )
```



```
plot(p , post.beta , type = 'l' , lwd = 1.5 , col = 'blue')
```



```
p.mom1 = delta.p * sum(p * post.beta)
p.mom2 = delta.p * sum(p^2 * post.beta)

print(p.mom1)
```

```
## [1] 0.1525424
```

```
print(p.mom2)
```

```
## [1] 0.0243555
```

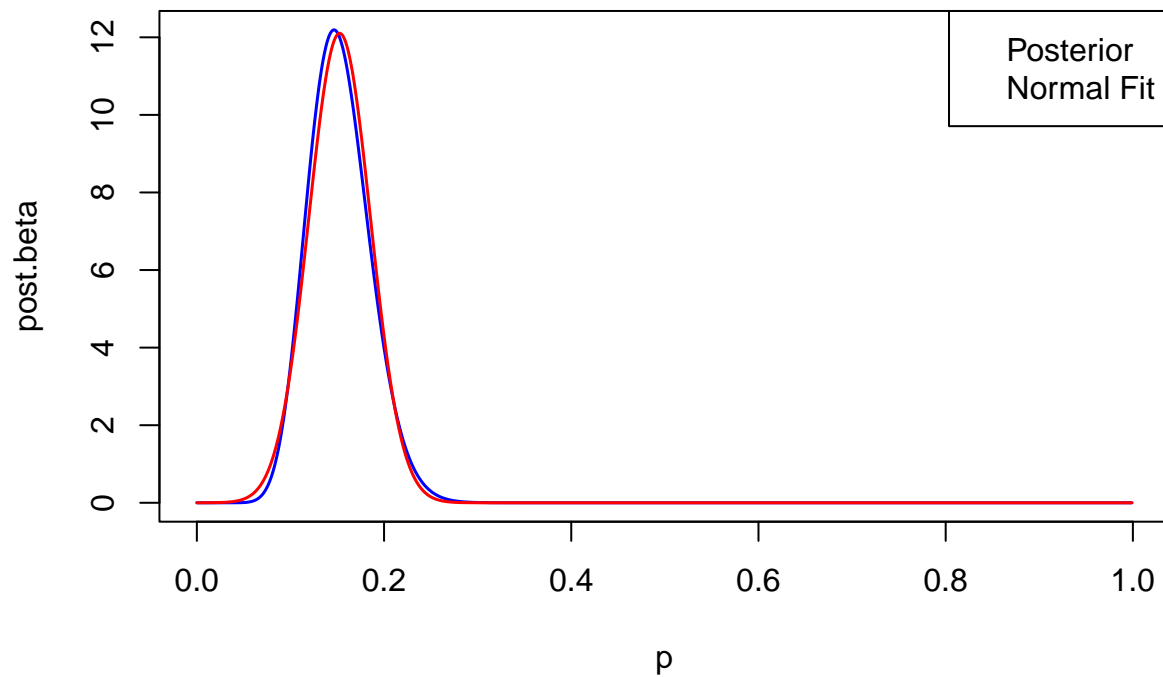
Using the two moments found above, we can fit a Gaussian to the posterior distribution:

```
m = p.mom1
s2 = p.mom2 - p.mom1 ** 2

p.normal = dnorm(p , m , sqrt(s2))
```

Now we plot the fit and the posterior together:

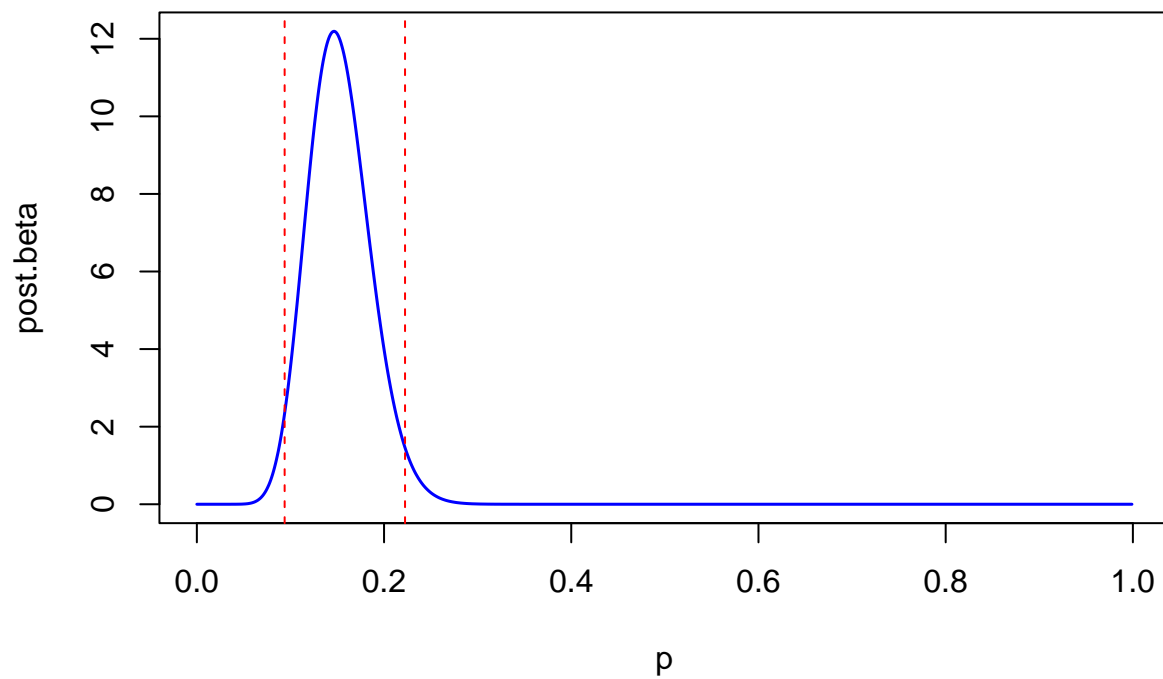
```
plot(p , post.beta , type = 'l' , lwd = 1.5 , col = 'blue')
lines(p , p.normal , type = 'l' , lwd = 1.5 , col = 'red')
legend('topright' , legend = c('Posterior', 'Normal Fit') , col = c('blue', 'red'))
```



```
ci = qbeta(c(0.025 , 0.975), alpha.prior + r , beta.prior + n - r)
ci
```

```
## [1] 0.09380263 0.22241633
```

```
plot(p , post.beta , type = 'l' , lwd = 1.5 , col = 'blue')
abline(v = ci[1], lty = "dashed", col = "red")
abline(v = ci[2], lty = "dashed", col = "red")
```

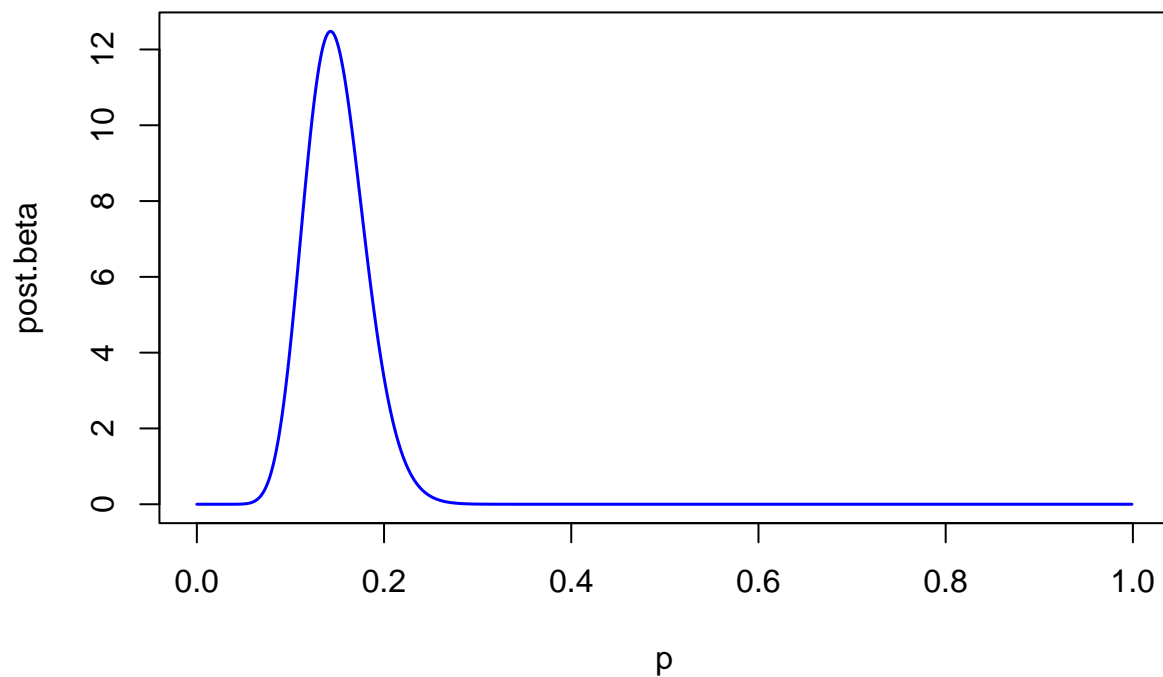


We repeat the same things for the beta prior:

```
alpha.prior = 1
beta.prior = 4

post.beta = dbeta(x = p , alpha.prior + r , beta.prior + n - r )

plot(p , post.beta , type = 'l' , lwd = 1.5 , col = 'blue')
```



```
p.mom1 = delta.p * sum(p * post.beta)
p.mom2 = delta.p * sum(p^2 * post.beta)

print(p.mom1)
```

```
## [1] 0.1487603
```

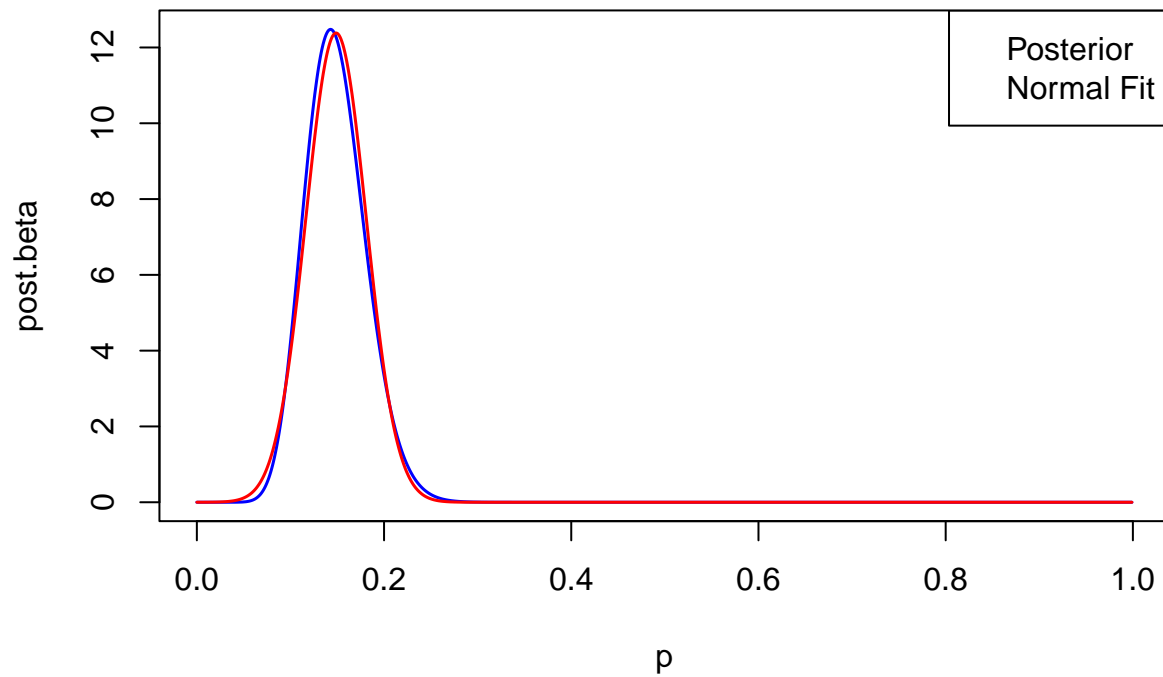
```
print(p.mom2)
```

```
## [1] 0.02316759
```

```
m = p.mom1
s2 = p.mom2 - p.mom1 ** 2

p.normal = dnorm(p , m , sqrt(s2))
```

```
plot(p , post.beta , type = 'l' , lwd = 1.5 , col = 'blue')
lines(p , p.normal , type = 'l' , lwd = 1.5 , col = 'red')
legend('topright' , legend = c('Posterior', 'Normal Fit') , col = c('blue', 'red'))
```



```
ci = qbeta(c(0.025 , 0.975), alpha.prior + r , beta.prior + n - r)
```

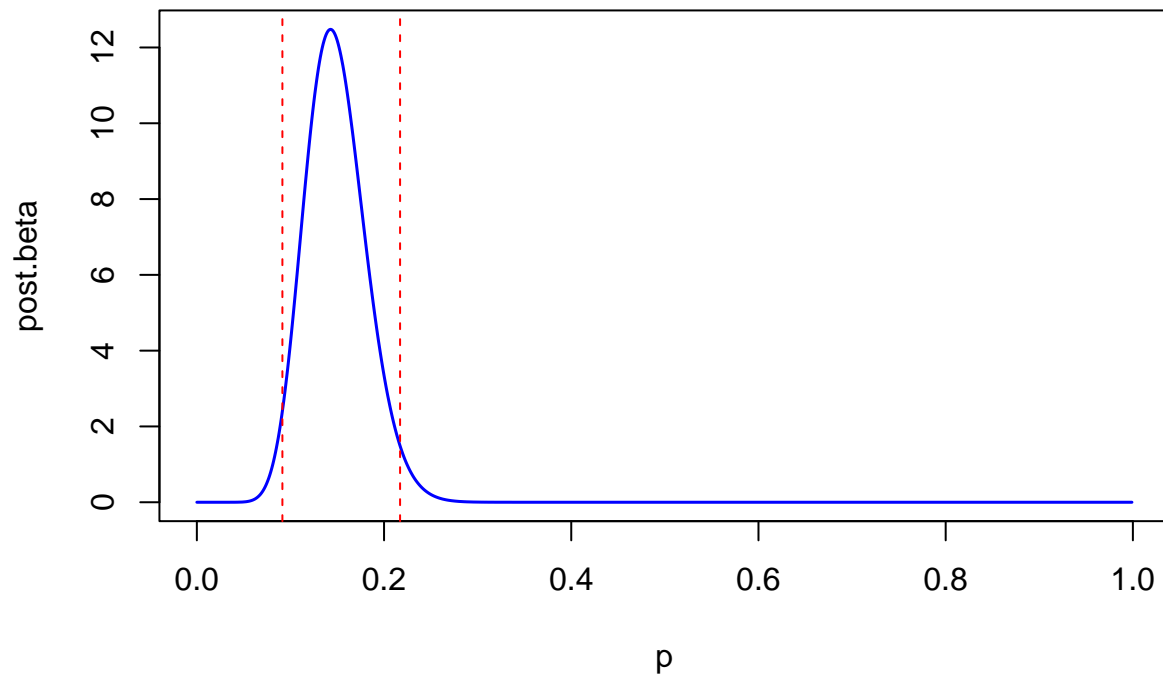
```
ci
```

```
## [1] 0.09138957 0.21710690
```

```
plot(p , post.beta , type = 'l' , lwd = 1.5 , col = 'blue')
```

```
abline(v = ci[1], lty = "dashed", col = "red")
```

```
abline(v = ci[2], lty = "dashed", col = "red")
```



Exercise 3:

a, b) In total there was 15 heads out of 30 trials. Again we assume a binomial model, and try flat and beta priors separately.

```
n = 30
r = 15
n.sample = 1000
delta.p = 1/n.sample

p = seq(from = 0 , by = delta.p , length.out = n.sample)
```

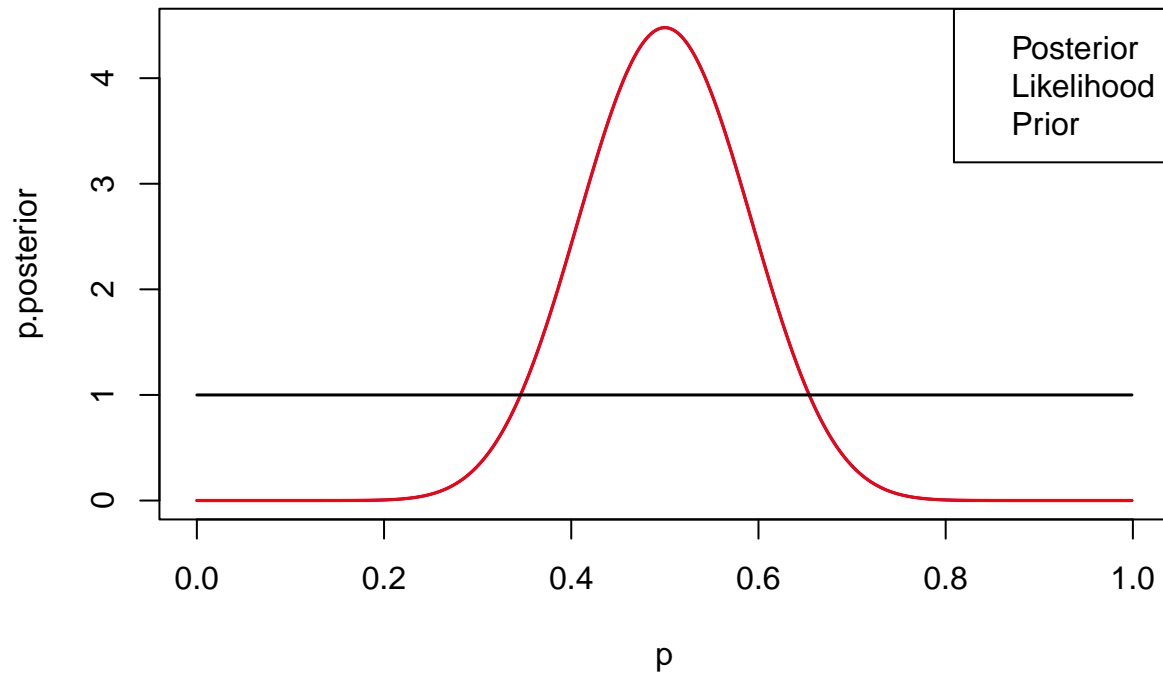
Assuming a uniform pdf we get:

```
uniform = function(x) 1
alpha.prior = 1
beta.prior = 1

p.prior = lapply(p , uniform)
likelihood = dbinom(x = r , size = n , prob = p)
likelihood = likelihood/(sum(likelihood) * delta.p)
p.posterior = dbeta(x = p , alpha.prior + r , beta.prior + n - r )
```

```
plot(p , p.posterior , type = 'l' , lwd = 1.5 , col = 'blue')
lines(p , likelihood , type = 'l' , lwd = 1.5 , col = 'red' )
lines(p , p.prior , type = 'l' , lwd = 1.5 , col = 'black')

legend('topright' , legend = c('Posterior' , 'Likelihood' , 'Prior') , col = c('blue' , 'red' , 'black'))
```



The most probable value for p can be obtained using the mode of the distribution:

```
mode = p[which.max(p.posterior)]
mode
```

```
## [1] 0.5
```

```
samples = sample(p , size = 1000 , replace = TRUE , prob = p.posterior)

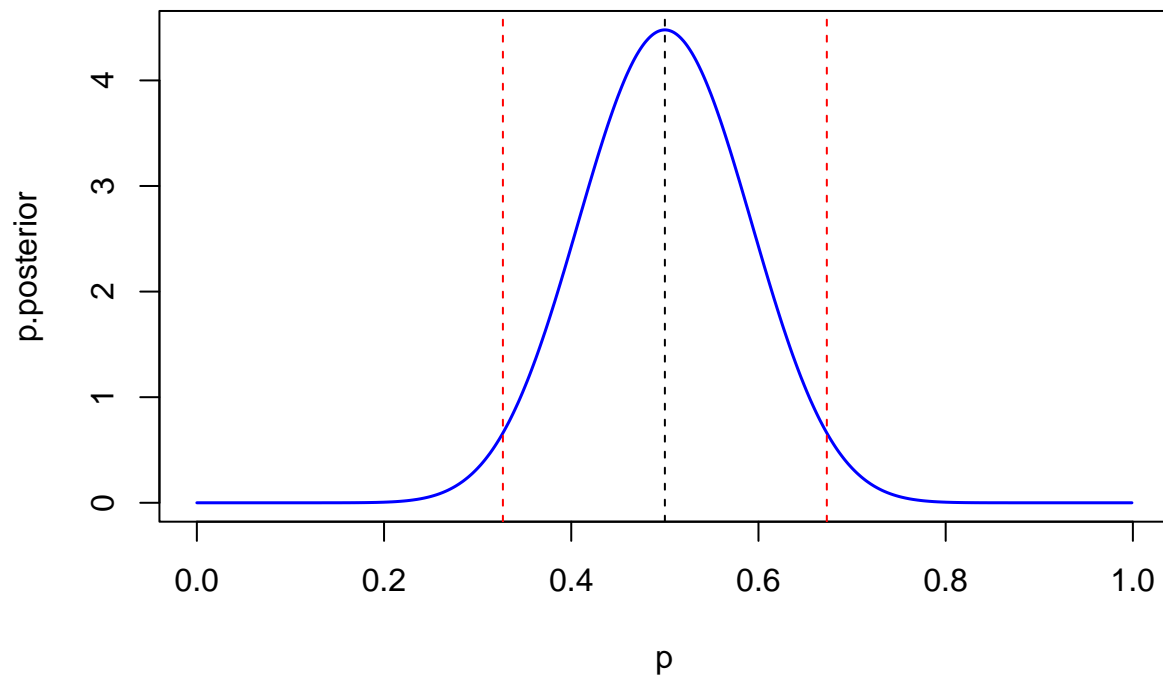
ci = quantile(samples , c(0.025 , 0.975))

ci
```

```
##      2.5%      97.5%
## 0.326975 0.673025
```

```
plot(p , p.posterior , type = 'l' , lwd = 1.5 , col = 'blue')
```

```
abline(v = mode, lty = "dashed", col = "black")
abline(v = ci[1], lty = "dashed", col = "red")
abline(v = ci[2], lty = "dashed", col = "red")
```



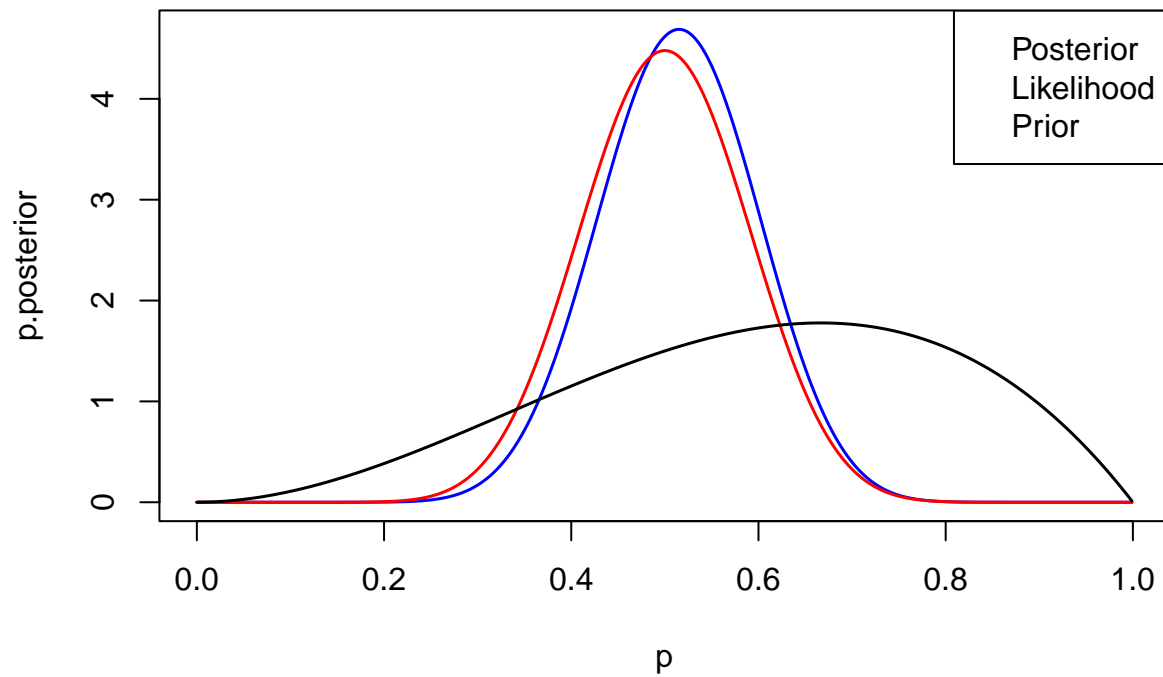
Assuming a Beta prior we get:

```
alpha.prior = 3
beta.prior = 2
```

```
p.prior = dbeta(x = p , alpha.prior , beta.prior)
likelihood = dbinom(x = r , size = n , prob = p)
likelihood = likelihood/(sum(likelihood) * delta.p)
p.posterior = dbeta(x = p , alpha.prior + r , beta.prior + n - r )
```

```
plot(p , p.posterior , type = 'l' , lwd = 1.5 , col = 'blue')
lines(p , likelihood , type = 'l' , lwd = 1.5 , col = 'red' )
lines(p , p.prior , type = 'l' , lwd = 1.5 , col = 'black')
```

```
legend('topright' , legend = c('Posterior' , 'Likelihood' , 'Prior') , col = c('blue' , 'red' , 'black'))
```

```
samples = sample(p , size = 1000 , replace = TRUE , prob = p.posterior)

ci = quantile(samples , c(0.025 , 0.975))

ci
```

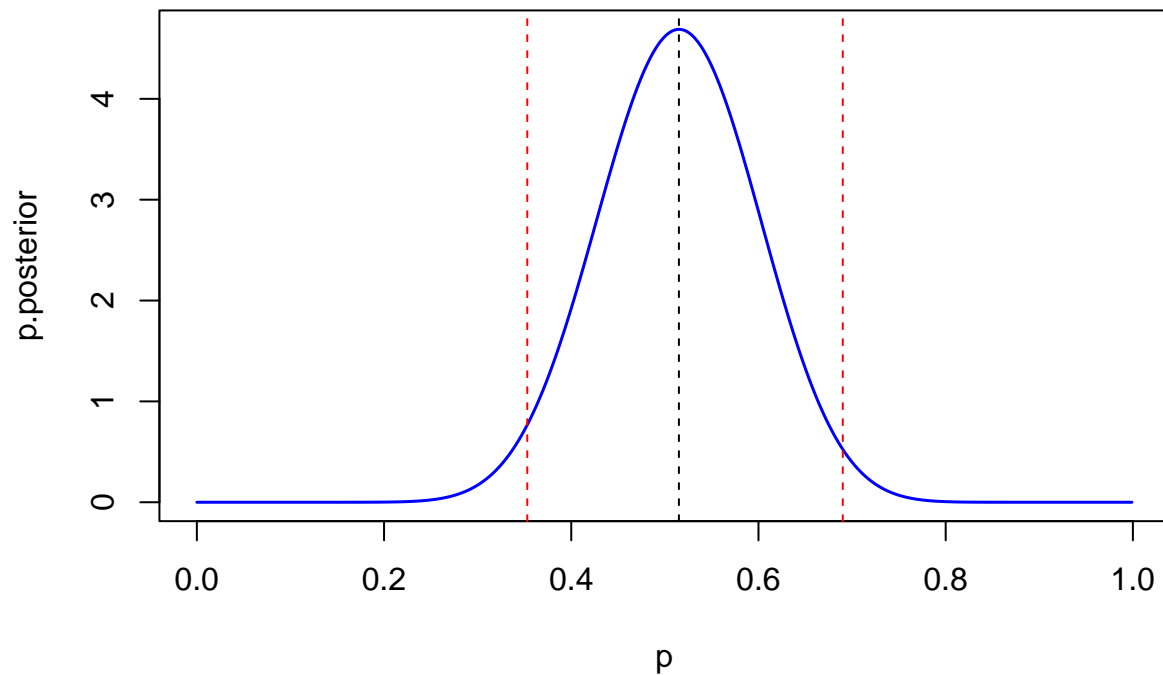
```
##      2.5%   97.5%
## 0.35300 0.69005
```

```
mode = p[which.max(p.posterior)]
mode
```

```
## [1] 0.515
```

```
plot(p , p.posterior , type = 'l' , lwd = 1.5 , col = 'blue')

abline(v = mode, lty = "dashed", col = "black")
abline(v = ci[1], lty = "dashed", col = "red")
abline(v = ci[2], lty = "dashed", col = "red")
```



c,d) Let's assume a uniform prior. We change the value of r and n according to the list of head and tails and compute the credibility value and the mode at each step.

```
uniform = function(x) 1  
alpha.prior = 1  
beta.prior = 1
```

```
r.list = c(0,0,0,0,0, 1, 1 , 1, 2, 3 , 3 , 3 , 4 , 5 , 6 , 6 , 7 , 7 , 8 , 8 , 9 , 10 , 10 , 11 , 11 , 12)
mode.list = c()
cimin.list = c()
cimax.list = c()

for(i in 1:30) {

  r = r.list[i]
  n = i

  p.prior = lapply(p , uniform)
  likelihood = dbinom(x = r , size = n , prob = p)
  likelihood = likelihood/(sum(likelihood) * delta.p)
  p.posterior = dbeta(x = p , alpha.prior + r , beta.prior + n - r )

  mode = p[which.max(p.posterior)]
  mode.list = c(mode.list , mode)
```

```

samples = sample(p , size = 1000 , replace = TRUE , prob = p.posterior)
ci = quantile(samples , c(0.025 , 0.975) , type = 2)

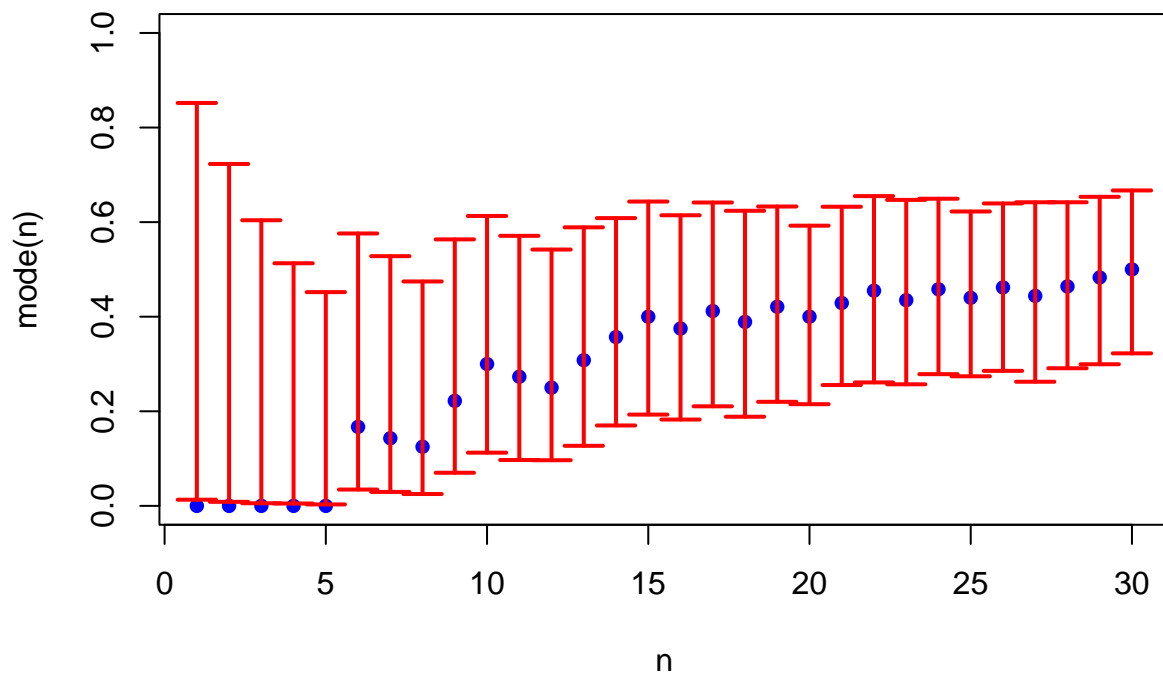
cimin.list = c(cimin.list , ci[[1]])
cimax.list = c( cimax.list , ci[[2]])
}

n.list <- 1:30

# plot scatter plot
plot(n.list, mode.list, pch=16, col="blue", xlab="n", ylim = c(0 ,1) , ylab="mode(n)")

# add error bars
arrows(n.list, cimin.list, n.list, cimax.list, angle=90, code=3, length=0.1, col="red", lwd=2, xpd=TRUE)

```



As we see, the modes and the credibility limits change as we include the results of more trials in the analysis.

Exercise 4:

We start by selecting a box randomly:

```

i = sample(0:5 , 1)
i

```

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

the probability of drawing a white ball from the i th box is:

```
p_w = i/5
p_b = 1 - p_w
```

so we do random sampling from the box; we create a list of samples with the length 60. Number 1 in the list represents white ball, and number 0 represents black.

```
num_trials = 60

samples = sample(c(0 , 1) , num_trials , replace = TRUE , prob = c(p_b , p_w))

samples

## [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [39] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Now we loop through the samples and box numbers and create a list of probabilities for each box at each step. Notice that at the beginning the prior probability for each box is uniform.

```
p_h = c(1/6 , 1/6 , 1/6 , 1/6 , 1/6 , 1/6)
ph_list = c()
ph_list[[1]] = p_h

for (j in 1:60){

  event = samples[j]

  if (event == 1) {p_E = c(0 , 1/5 , 2/5 , 3/5 , 4/5 , 5/5)}
  else {p_E = c(5/5 , 4/5 , 3/5 , 2/5 , 1/5 , 0/5)}

  p_h = p_E * p_h / (sum(p_E * p_h))

  ph_list[[j + 1]] = p_h

}
```

```
# Convert ph_list to a dataframe
ph_df <- do.call(rbind, ph_list)

# Set the iteration numbers as row names
rownames(ph_df) <- 0:60

# Set the layout for the plots
par(mfrow = c(3, 2))

# Adjust the figure margins
par(mar = c(4, 4, 2, 1))

# Create separate plots for each box
for (i in 1:6) {
```

```

box_probs <- ph_df[, i]
box_name <- paste("Box", i)

# Plot the probabilities for the current box as scatter plot
plot(rownames(ph_df), box_probs, type = "p", col = i, pch = 16, xlab = "Iteration", ylab = "Probability")
}

```

