Lab 1: Mixture of Beta Priors

Author: Shang-Chieh (Jay), Wei

Total Grade for Lab 1: /15

Comments (optional)

Template for lab report

Instructions: This is the template you will use to type up your responses to the exercises. To produce a document that you can print out and turn in just click on Knit PDF above. All you need to do to complete the lab is to type up your BRIEF answers and the R code (when necessary) in the spaces provided below.

It is strongly recommended that you knit your document regularly (minimally after answering each exercise) for two reasons.

1. Ensure that there are no errors in your code that would prevent the document from knitting.
2. View the instructions and your answers in a more legible, attractive format.

```
# Any text BOTH preceded by a hashtag AND within the ```{r} ``` code chunk is a comment.
# R indicates a comment by turning the text green in the editor, and brown in the knitted
# document.
# Comments are not treated as a command to be interpreted by the computer.
# They normally (briefly!) describe the purpose of your command or chunk in plain English.
# However, for this class, they will have a different goal, as the text above and below
# each chunk should sufficiently describe the chunk's contents.
# For this class, comments will be used to indicate where your code should go, or to give
# hints for what the code should look like.
```

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
require(gridExtra)
```

```
## Loading required package: gridExtra
```

```
require(ProbBayes)
```

```
## Loading required package: ProbBayes
```

```
## Loading required package: LearnBayes
```

```
## Loading required package: shiny
```

```r
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.1     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::combine() masks gridExtra::combine()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

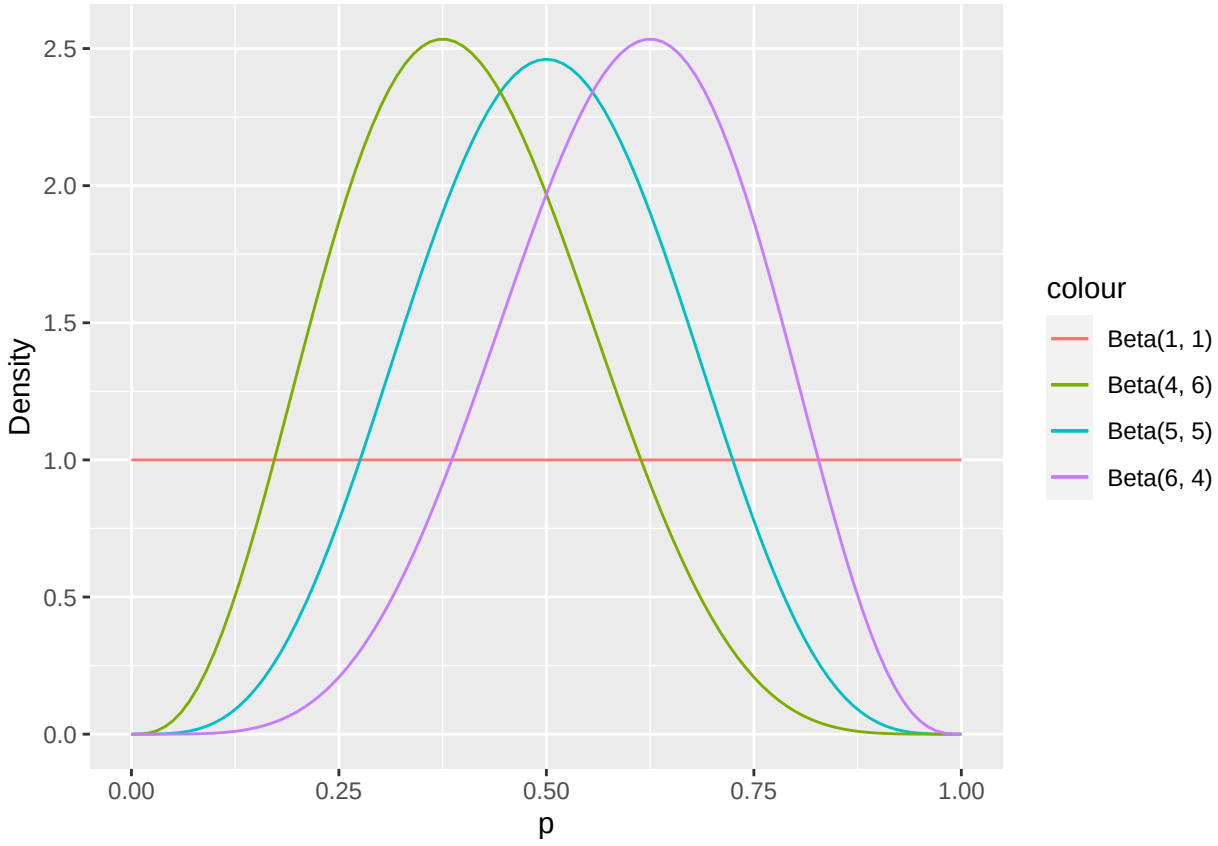
```r
crcblue <- "#2905a1"
```

Mixture of Beta Priors

Estimate the probability p of teen recidivism based on a study in which there were $n = 43$ individuals released from incarceration and $y = 15$ re-offenders within 36 months.

```r
# If you use the exact solution...
# Here is some sample script to plot multiple Beta densities on the same graph
# Delete "eval=FALSE" above to see output

require(ggplot2)

ggplot(data = data.frame(p = c(0, 1)), aes(p)) +
  stat_function(fun = dbeta, args = list(shape1 = 1, shape2 = 1), aes(color = "Beta(1, 1)")) +
  stat_function(fun = dbeta, args = list(shape1 = 5, shape2 = 5), aes(color = "Beta(5, 5)")) +
  stat_function(fun = dbeta, args = list(shape1 = 4, shape2 = 6), aes(color = "Beta(4, 6)")) +
  stat_function(fun = dbeta, args = list(shape1 = 6, shape2 = 4), aes(color = "Beta(6, 4)")) +
  ylab("Density")
```

Exercise 1: **Using a Beta$(2, 8)$ prior for p, plot the prior $\pi(p)$ and the posterior $\pi(p \mid y)$ as functions of p. Find the posterior mean and standard deviation of p. Find a 95% quantile-based credible interval. You can use either the exact solution or approximation through Monte Carlo sim-**



ulation.

```
# If you use approximation through Monte Carlo simulation...
# Here is some sample script to generate Beta samples and
# plot multiple Beta densities on the same graph
# Delete "eval=FALSE" above to see output
require(reshape2)
```

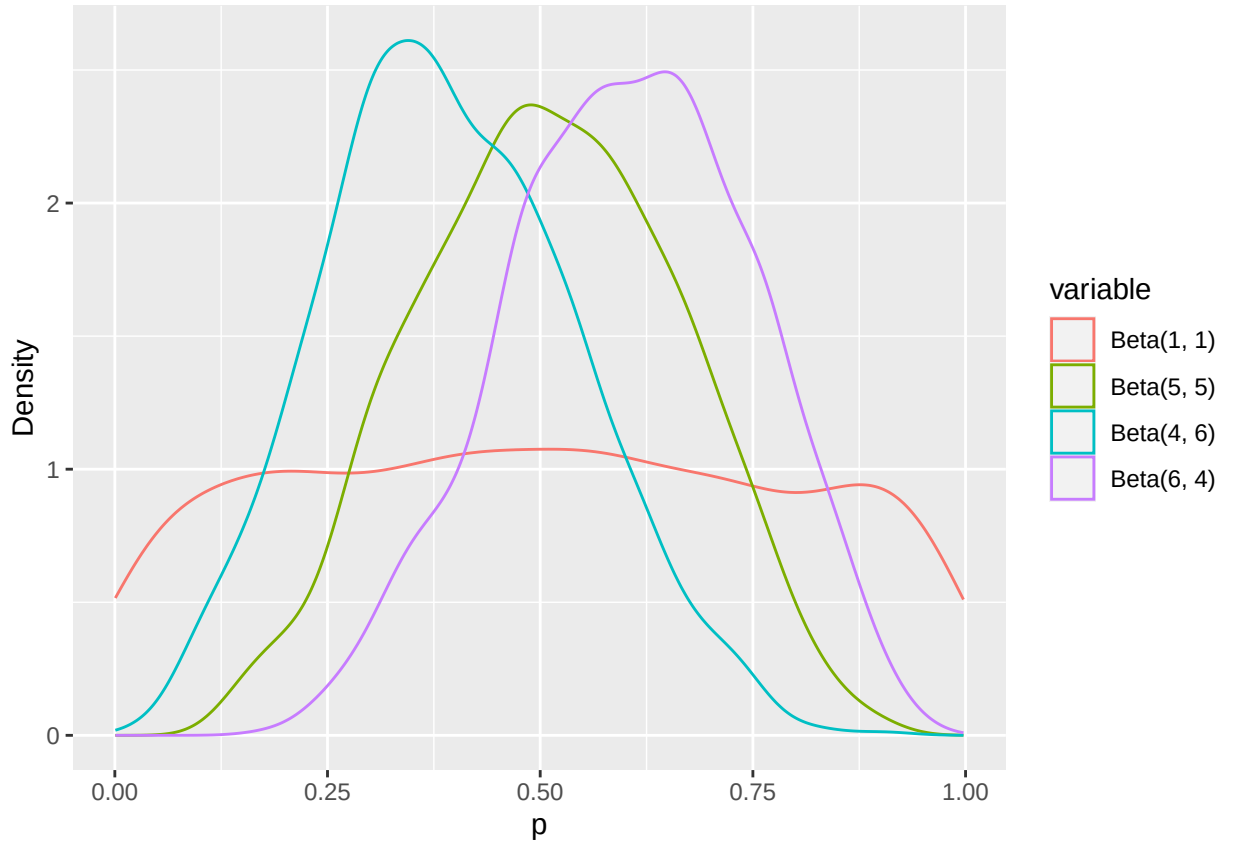
```
## Loading required package: reshape2
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
require(ggplot2)

set.seed(123)
S <- 1000
Beta11samples <- rbeta(S, shape1 = 1, shape2 = 1)
Beta55samples <- rbeta(S, shape1 = 5, shape2 = 5)
Beta46samples <- rbeta(S, shape1 = 4, shape2 = 6)
Beta64samples <- rbeta(S, shape1 = 6, shape2 = 4)
df <- as.data.frame(cbind(seq(1:S), Beta11samples, Beta55samples, Beta46samples, Beta64samples))
```

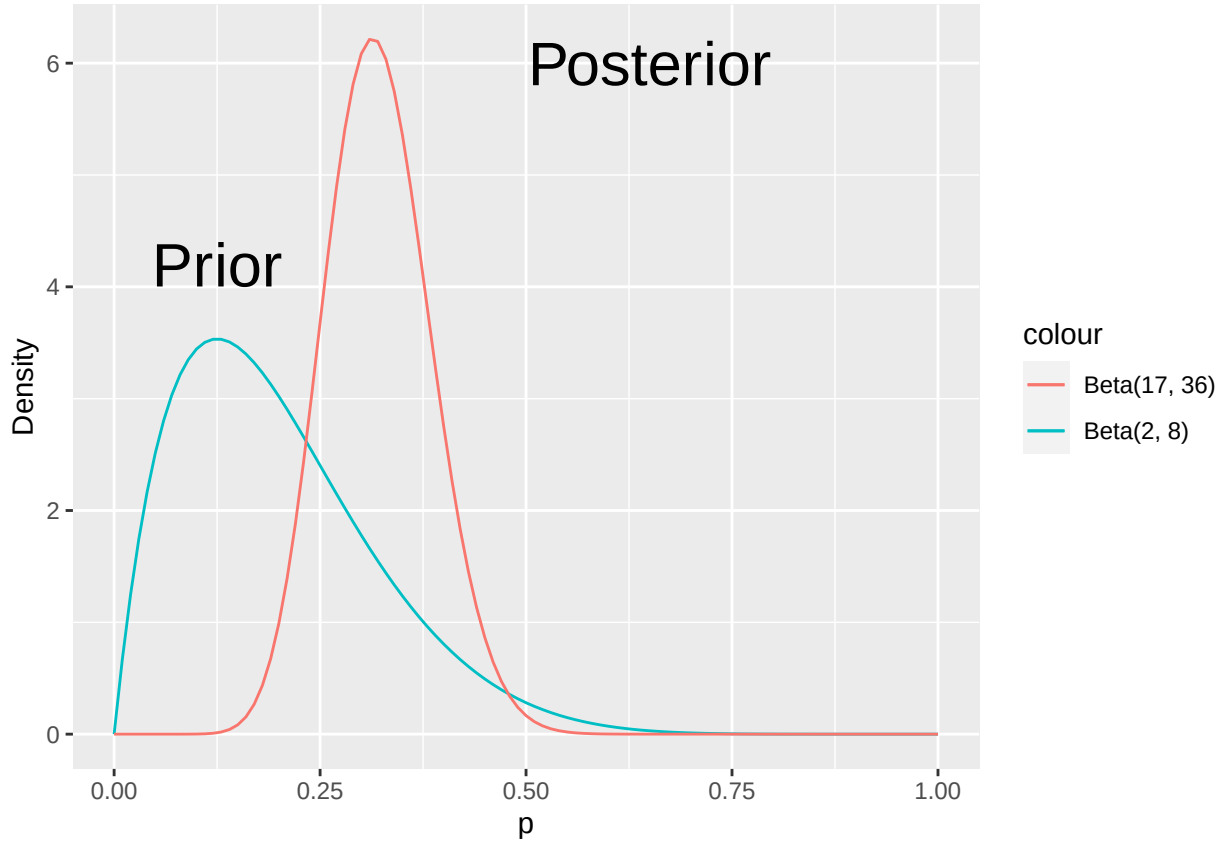
```
names(df) <- c("Index", "Beta(1, 1)", "Beta(5, 5)", "Beta(4, 6)", "Beta(6, 4)")
df_long <- melt(df, id = "Index")

ggplot(data = df_long, aes(value, colour = variable)) +
  geom_density() +
  xlab("p") + ylab("Density")
```



Below is the solutions.

We first show analytic solutions.

```
# exact solution
n <- 43
y <- 15

require(ggplot2)

ggplot(data = data.frame(p = c(0, 1)), aes(p)) +
  stat_function(fun = dbeta, args = list(shape1 = 2, shape2 = 8), aes(color = "Beta(2, 8)")) +
  stat_function(fun = dbeta, args=list(shape1 = 2+y, shape2= 8+n-y), aes(color = "Beta(17, 36)")) +
  annotate(geom = "text", x = .125, y = 4.2,
           size = 8, label = "Prior") +
  annotate(geom = "text", x = .65, y = 6,
           size = 8, label="Posterior") +
  ylab("Density")
```



Recall: for a Beta(a,b), the mean is $\frac{a}{a+b}$, and the variance is $\frac{ab}{(a+b+1)(a+b)^2}$

```
# calculate analytic solution of mean and std.dev

a=2+y
b=8+n-y
cat("The posterior mean of p is ", a/(a+b), sep="", "\n")
```

```
## The posterior mean of p is 0.3207547
```

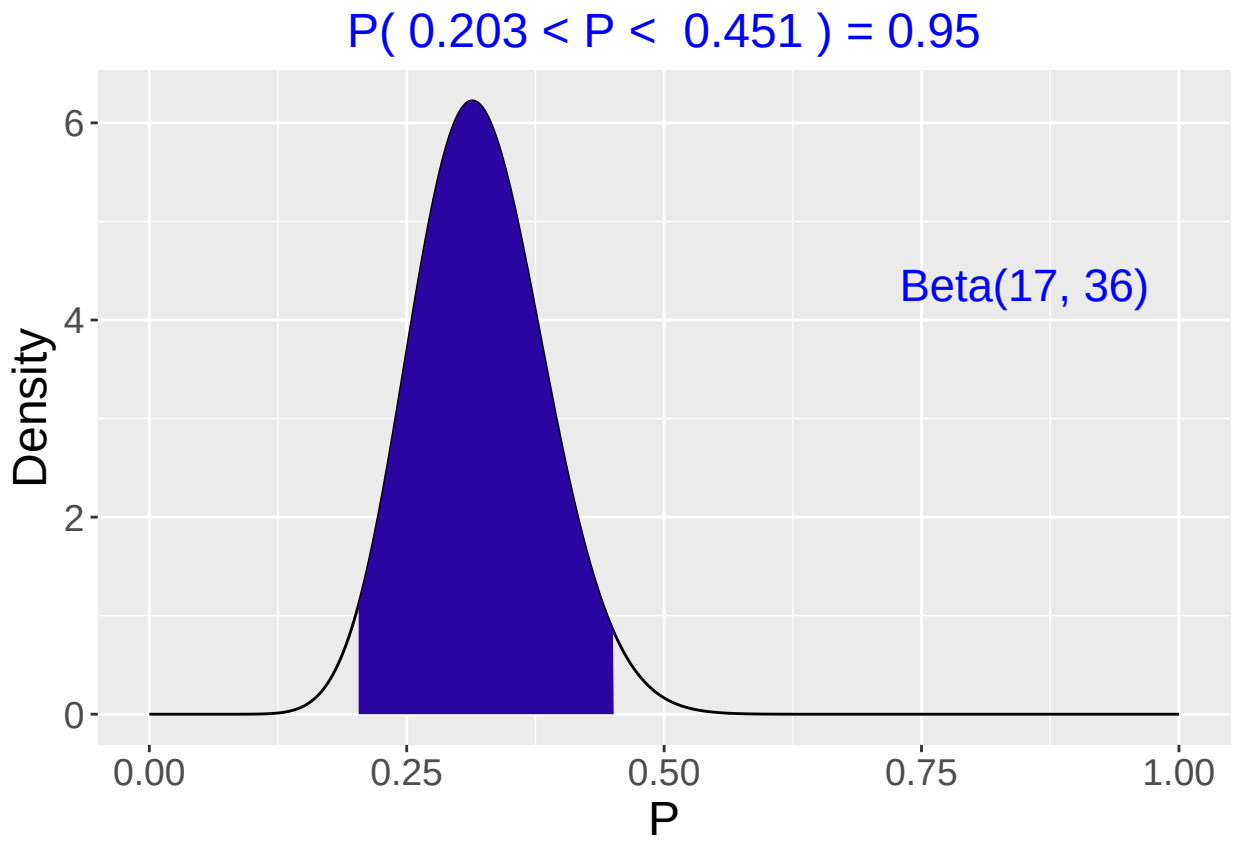
```
cat("The posterior std.deviation of p is ", sqrt(a*b/((a+b+1)*(a+b)^2)), sep="", "\n")
```

```
## The posterior std.deviation of p is 0.0635189
```

```
# calculate 95% quantile-based credible interval

c(qbeta(0.025, a, b), qbeta(0.975, a, b))
```

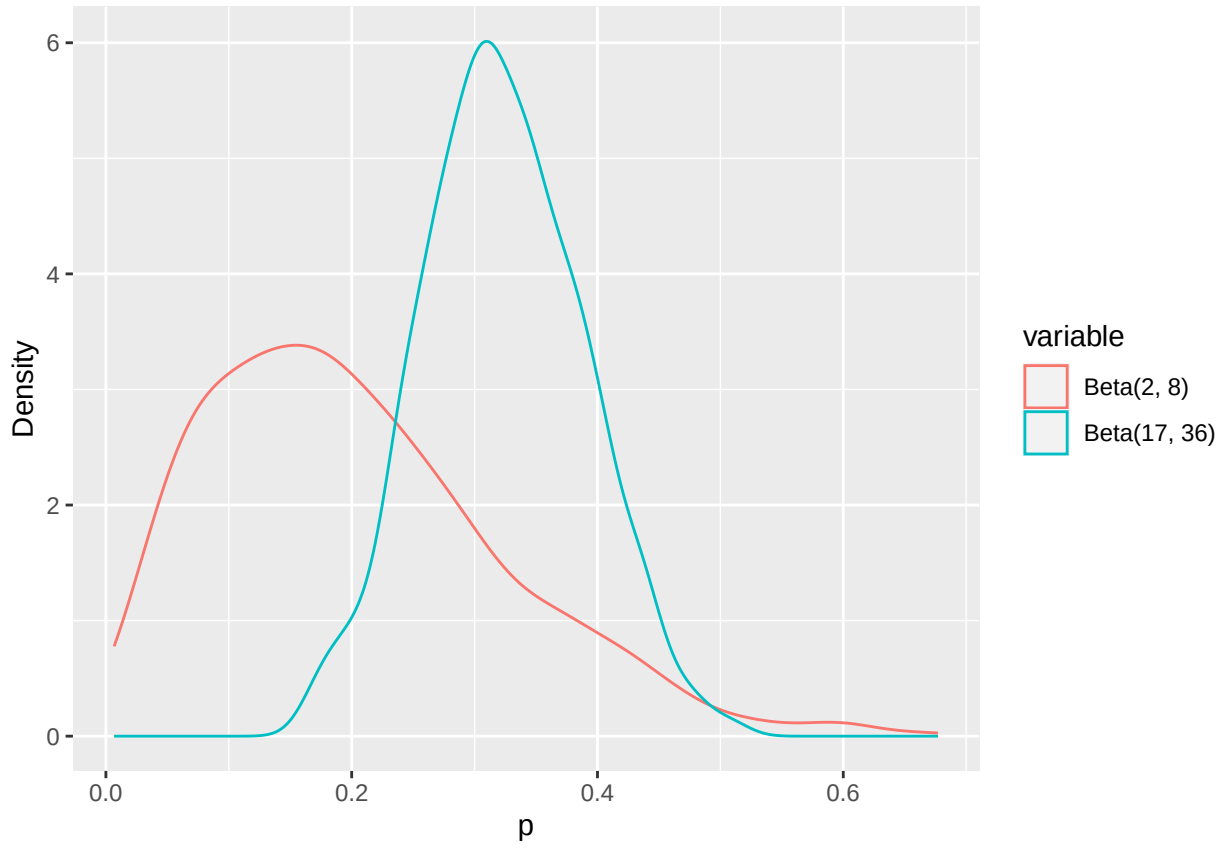
```
## [1] 0.2032978 0.4510240
```

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



Now, we show approximated solutions via Monte Carlo simulation.

```
# approximation through Monte Carlo simulation

require(reshape2)
require(ggplot2)
set.seed(123)
S <- 1000
Beta28samples <- rbeta(S, shape1 = 2, shape2 = 8)
Betaabsamples <- rbeta(S, shape1 = a, shape2 = b)
df <- as.data.frame(cbind(seq(1:S), Beta28samples, Betaabsamples))
names(df) <- c("Index", "Beta(2, 8)", "Beta(17, 36)")
df_long <- melt(df, id = "Index")

ggplot(data = df_long, aes(value, colour = variable)) +
  geom_density() +
  xlab("p") + ylab("Density")
```




```
# The approximation is not bad!!
mean(Betaabsamples)
```

```
## [1] 0.3225574
```

```
sd(Betaabsamples)
```

```
## [1] 0.06486996
```

```
# 95% credible interval
quantile(Betaabsamples, c(0.025, 0.975))
```

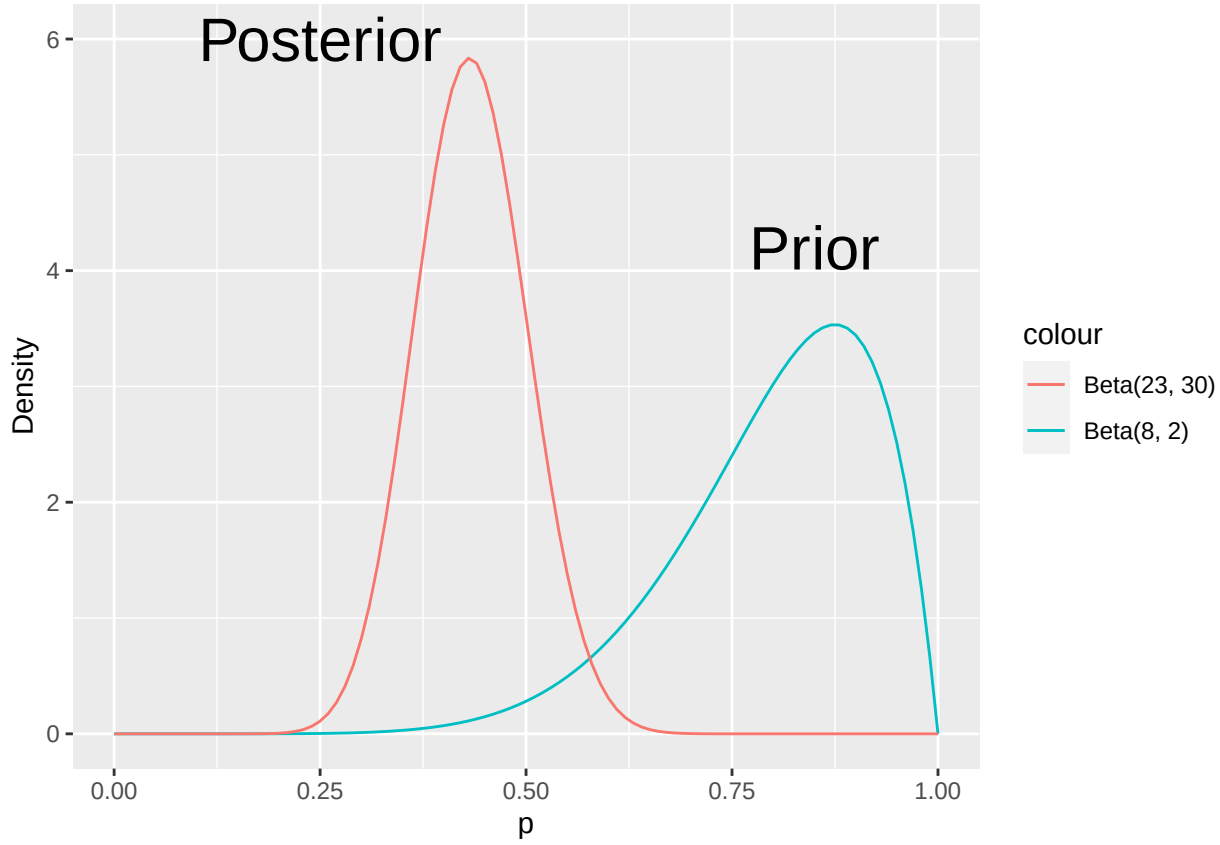
```
##      2.5%     97.5%
## 0.1963694 0.4489108
```

Grade for Exercise 1: /5

Comments:

Exercise 2: **Repeat Exercise 1, but using a Beta$(8, 2)$ prior for p.** We first show analytic solutions.

```r
# exact solution
n <- 43
y <- 15

require(ggplot2)

ggplot(data = data.frame(p = c(0, 1)), aes(p)) +
  stat_function(fun = dbeta, args = list(shape1 = 8, shape2 = 2), aes(color = "Beta(8, 2)")) +
  stat_function(fun = dbeta, args=list(shape1 = 8+y, shape2= 2+n-y), aes(color = "Beta(23, 30)")) +
  annotate(geom = "text", x = .85, y = 4.2,
           size = 8, label = "Prior") +
  annotate(geom = "text", x = .25, y = 6,
           size = 8, label="Posterior") +
  ylab("Density")
```




```r
# calculate analytic solution of mean and std.dev

a=8+y
b=2+n-y
cat("The posterior mean of p is ", a/(a+b), sep="", "\n")
```

```
## The posterior mean of p is 0.4339623
```

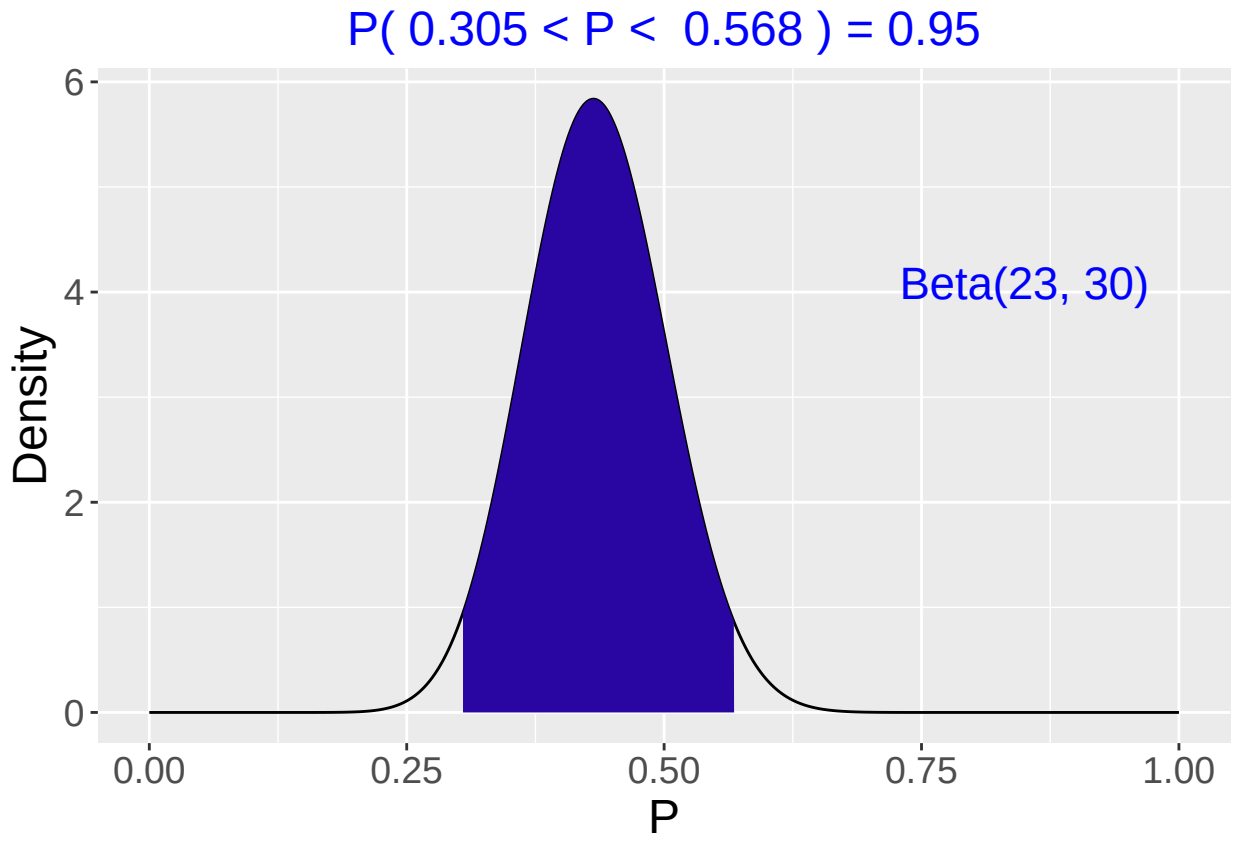
```r
cat("The posterior std.deviation of p is ", sqrt(a*b/((a+b+1)*(a+b)^2)), sep="", "\n")
```

```
## The posterior std.deviation of p is 0.06744532
```

```r
# calculate 95% quantile-based credible interval

c(qbeta(0.025, a, b), qbeta(0.975, a, b))
```

```
## [1] 0.3046956 0.5679528
```

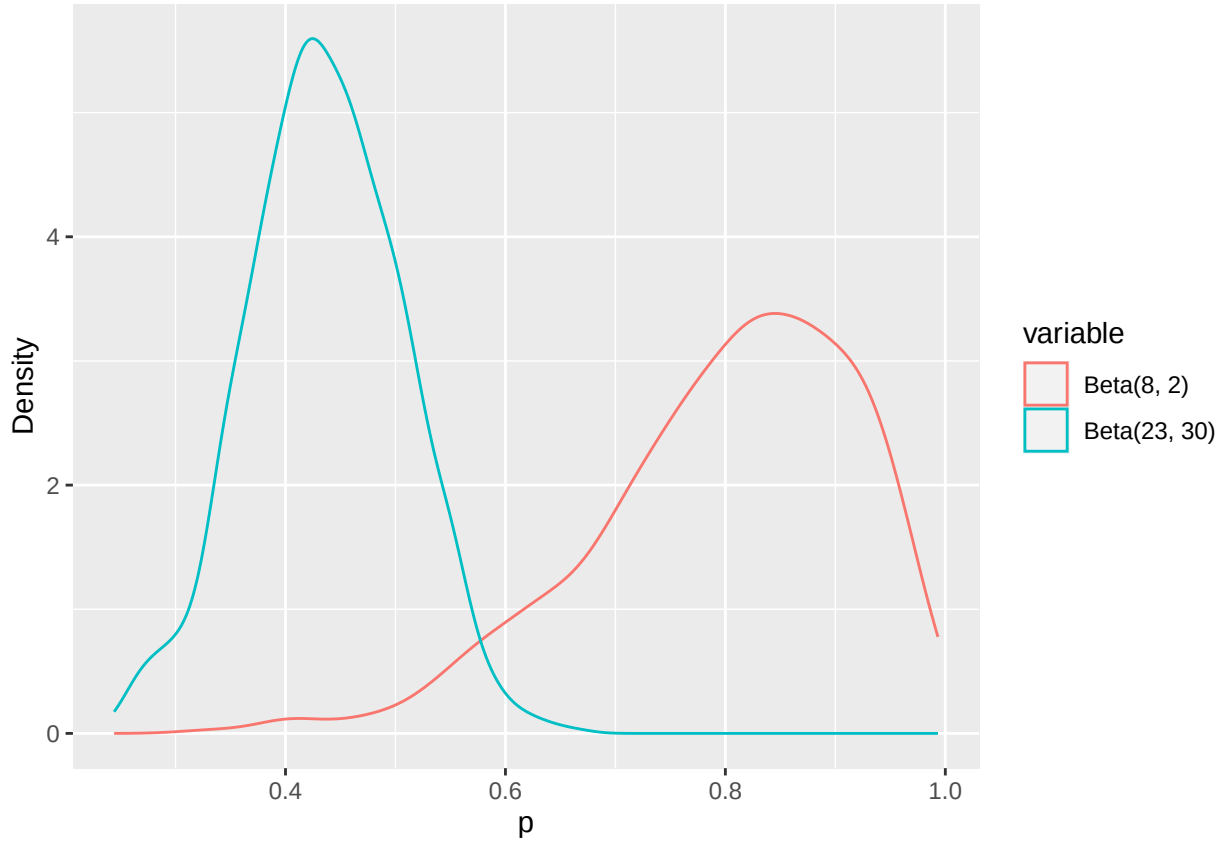
```r
beta_interval(0.95, c(a, b), Color= crcblue) +
  theme(text=element_text(size=18))
```



Now, we show approximated solutions via Monte Carlo simulation.

```r
# approximation through Monte Carlo simulation

require(reshape2)
require(ggplot2)
set.seed(123)
S <- 1000
Beta82samples <- rbeta(S, shape1 = 8, shape2 = 2)
Betaabsamples <- rbeta(S, shape1 = a, shape2 = b)
df <- as.data.frame(cbind(seq(1:S), Beta82samples, Betaabsamples))
```

```
names(df) <- c("Index", "Beta(8, 2)", "Beta(23, 30)")
df_long <- melt(df, id = "Index")

ggplot(data = df_long, aes(value, colour = variable)) +
  geom_density() +
  xlab("p") + ylab("Density")
```




```
# The approximation is not bad!!
mean(Betaabsamples)
```

```
## [1] 0.4360609
```

```
sd(Betaabsamples)
```

```
## [1] 0.06947008
```

```
# 95% credible interval
quantile(Betaabsamples, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.2945910 0.5662667
```

Grade for Exercise 2: /5

Comments:

Exercise 3: **Consider the following prior distribution for p, a $75-25\%$ mixture of a Beta$(2,8)$ and a Beta$(8,2)$ prior distribution. Plot this prior distribution and compare it to the priors in Exercise 1 and Exercise 2. Describe what sort of prior opinion this may represent.**

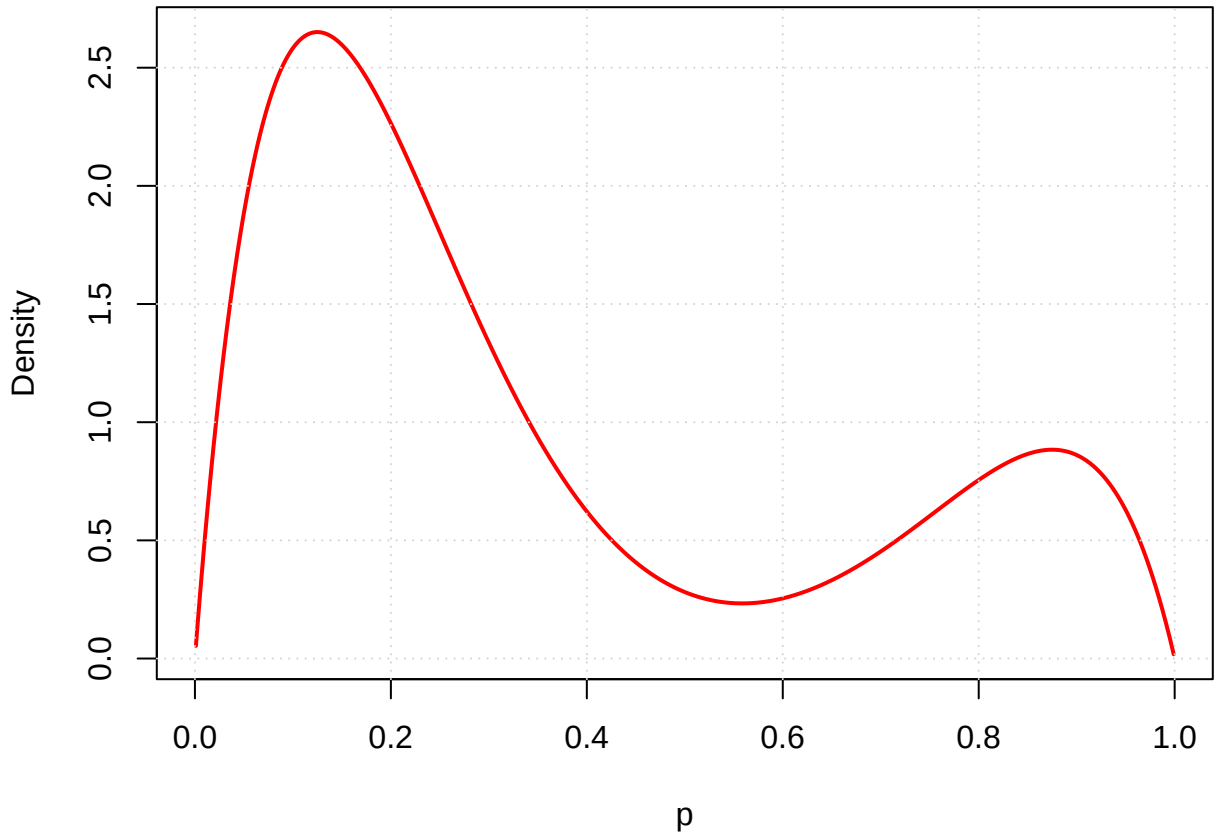
$$\pi(p) = \frac{1}{4}\frac{\Gamma(10)}{\Gamma(2)\Gamma(8)}[3p(1-p)^7 + p^7(1-p)],$$

```
# If you use the exact solution...
# Here is some sample script to create density of a 75%-25% mixture Beta prior
# Delete "eval=FALSE" above to see output

p <- seq(0.001, 0.999, length = 1000)
mixture <- 0.75*dbeta(p, 1, 1) + 0.25*dbeta(p, 5, 5)
```

```
# If you use approximation through Monte Carlo simulation...
# Here is some sample script to generate draws of a 75%-25% mixture Beta prior
# Delete "eval=FALSE" above to see output

set.seed(123)
S <- 1000
MixtureBetaSamples <- rep(NA, S)
for (s in 1:S){
  component <- rbinom(1, 1, 0.75) # this is bernoulli
  if (component == 1){
    MixtureBetaSamples[s] <- rbeta(1, shape1 = 1, shape2 = 1)
  }
  else
    MixtureBetaSamples[s] <- rbeta(1, shape1 = 5, shape2 = 5)
}
```

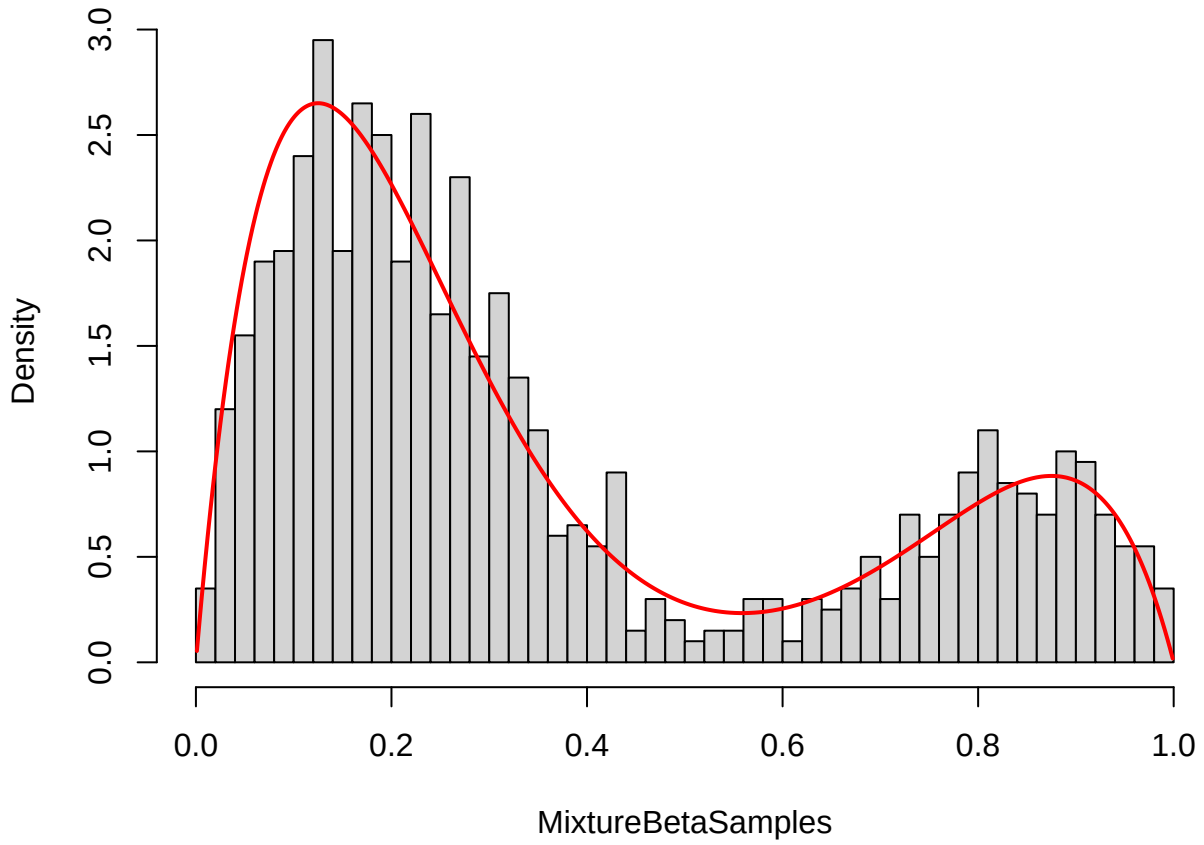
We first show analytic solutions.

```
p <- seq(0.001, 0.999, length = 1000)
mixture <- 0.75*dbeta(p, 2, 8) + 0.25*dbeta(p, 8, 2)


par(mai=c(0.9, 0.9, 0.1, 0.1))
plot(x= p, y= mixture, type = "l", col= "red", lwd = 2,
     ylab ="Density")
grid()
```



Now, we show approximated solutions via Monte Carlo simulation.

```
set.seed(123)
S <- 1000
MixtureBetaSamples <- rep(NA, S)
for (s in 1:S){
  component <- rbinom(1, 1, 0.75)
  if (component == 1){
    MixtureBetaSamples[s] <- rbeta(1, shape1 = 2, shape2 = 8)
  }
  else
    MixtureBetaSamples[s] <- rbeta(1, shape1 = 8, shape2 = 2)
}

par(mai=c(0.9, 0.9, 0.1, 0.1))
hist(MixtureBetaSamples, main=NULL, prob=TRUE, breaks=50)
lines(p, 0.75*dbeta(p, 2, 8) + 0.25*dbeta(p, 8, 2), col="red", lwd =2)
```



This prior suggests that the officers believe values of p (probability of teen recidivism) can be either high or low.

In particular they have stronger belief that the values of p will be small.

Grade for Exercise 3: /5

Comments: