Bayesian Inference for a Proportion (R scripts)

Jingchen (Monika) Hu

MATH 347 Bayesian Statistics

Installing the necessary packages

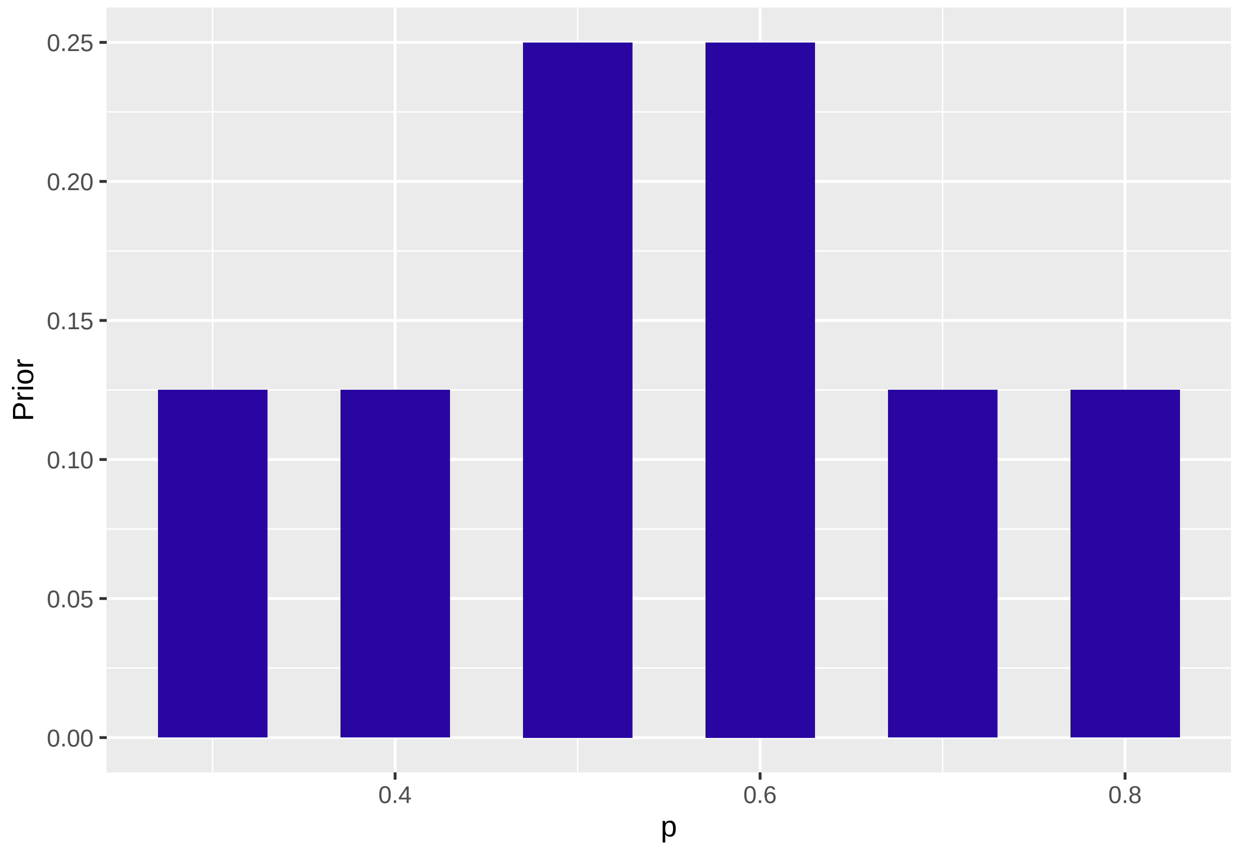
```
install.packages("devtools")
require(devtools)
devtools::install_github("bayesball/ProbBayes")

require(ggplot2)
require(gridExtra)
require(ProbBayes)
require(tidyverse)
crcblue <- "#2905a1"
```

Example: Tokyo Express customers' dining preference

Bayesian inference with discrete priors

Using R/RStudio to express and plot the prior $\pi_{owner}(p)$

```
bayes_table <- data.frame(p = seq(.3, .8, by=.1),
                          Prior = c(0.125, 0.125, 0.250,
                                    0.250, 0.125, 0.125))
ggplot(data=bayes_table, aes(x=p, y=Prior)) +
  geom_bar(stat="identity", fill=crcblue, width = 0.06)
```



Use R/RStudio to compute the likelihood function

```
bayes_table$Likelihood <- dbinom(12, size=20,
                                 prob=bayes_table$p)
bayes_table
```

```
##     p Prior  Likelihood
## 1 0.3 0.125 0.003859282
## 2 0.4 0.125 0.035497440
## 3 0.5 0.250 0.120134354
## 4 0.6 0.250 0.179705788
## 5 0.7 0.125 0.114396740
## 6 0.8 0.125 0.022160877
```

Use R/RStudio to compute and plot the posterior

```
bayesian_crank(bayes_table) -> bayes_table
bayes_table
```

```
##     p Prior  Likelihood      Product   Posterior
## 1 0.3 0.125 0.003859282 0.0004824102 0.004975901
## 2 0.4 0.125 0.035497440 0.0044371799 0.045768032
## 3 0.5 0.250 0.120134354 0.0300335884 0.309786454
## 4 0.6 0.250 0.179705788 0.0449264469 0.463401326
## 5 0.7 0.125 0.114396740 0.0142995925 0.147495530
## 6 0.8 0.125 0.022160877 0.0027701096 0.028572757
```

```
# check the above result

bayes_table$Product[1]/sum(bayes_table$Product)
```

```
## [1] 0.004975901
```

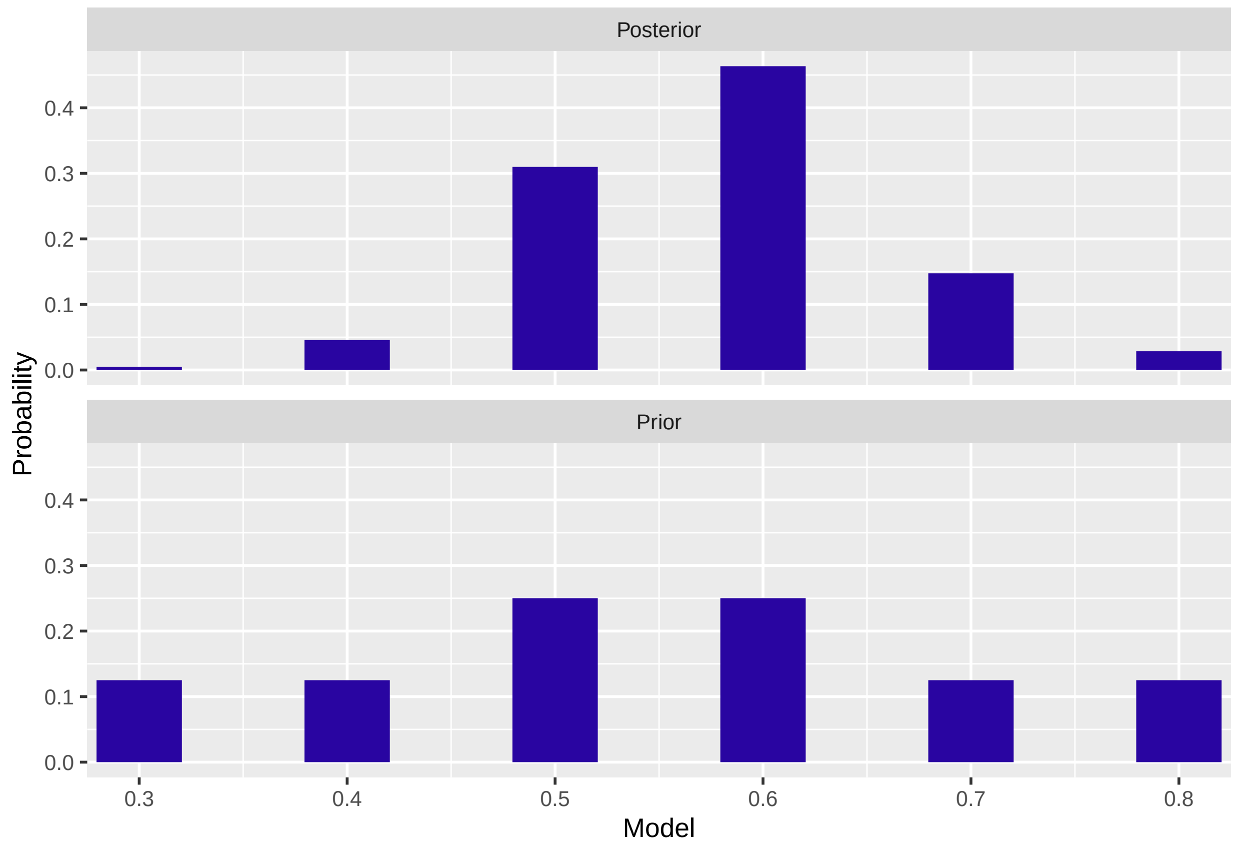
```
bayesian_crank(bayes_table) -> bayes_table
sum(bayes_table$Posterior[bayes_table$p > 0.5])
```

```
## [1] 0.6394696
```

```
sum(bayes_table$Posterior[bayes_table$p < 0.4])
```

```
## [1] 0.004975901
```

```
prior_post_plot(bayes_table, Color = crcblue) +
  theme(text=element_text(size=10))
```



Continuous priors - the Beta distribution

Step 1: Prior distribution

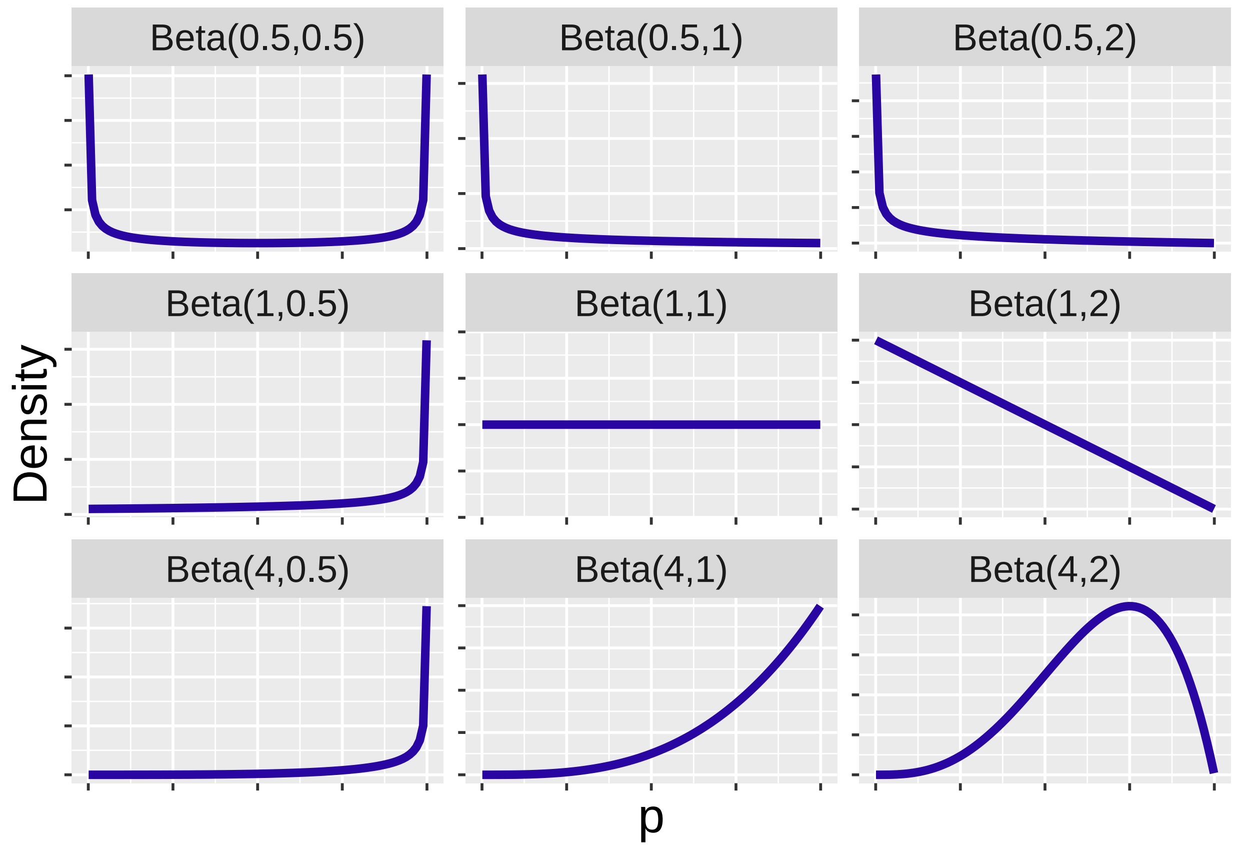
```
bayes_table
```

```
##     p Prior  Likelihood      Product   Posterior
## 1 0.3 0.125 0.003859282 0.0004824102 0.004975901
## 2 0.4 0.125 0.035497440 0.0044371799 0.045768032
## 3 0.5 0.250 0.120134354 0.0300335884 0.309786454
## 4 0.6 0.250 0.179705788 0.0449264469 0.463401326
## 5 0.7 0.125 0.114396740 0.0142995925 0.147495530
## 6 0.8 0.125 0.022160877 0.0027701096 0.028572757
```

Examples of Beta curves

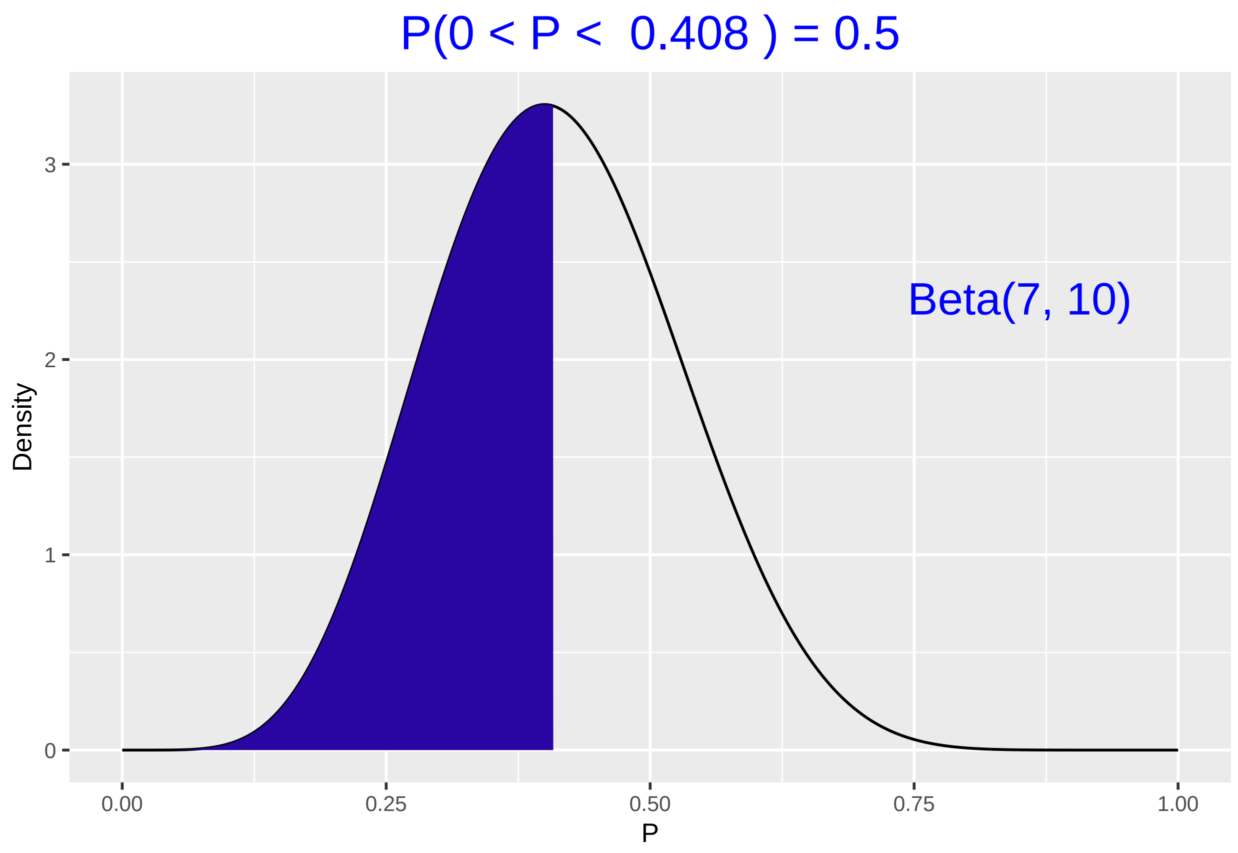
```
betapars <- matrix(c(0.5, 0.5,
                     0.5, 1,
                     0.5, 2,
                     1, 0.5,
                     1, 1,
                     1, 2,
                     4, 0.5,
                     4, 1,
                     4, 2),
                     9, 2, byrow = TRUE)
p <- seq(.001, .999, length.out = 100)
BETA <- NULL
for (j in 1:9){
  df <- data.frame(p = p, Density = dbeta(p,
                    betapars[j, 1], betapars[j, 2]))
  df$Type <-  paste("Beta(", betapars[j, 1],
                    ",", betapars[j, 2], ")",
                    sep = "")
  BETA <- rbind(BETA, df)
}
ggplot(BETA, aes(p, Density)) +
  geom_line(color = crcblue, size = 1.5) +
  facet_wrap(~ Type, scale = "free") +
  increasefont() +
  theme(axis.text=element_blank())
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



Choose a Beta curve to represent prior opinion

```
beta_quantile(0.5, c(7, 10), Color = crcblue) +
  theme(text=element_text(size=10))
```

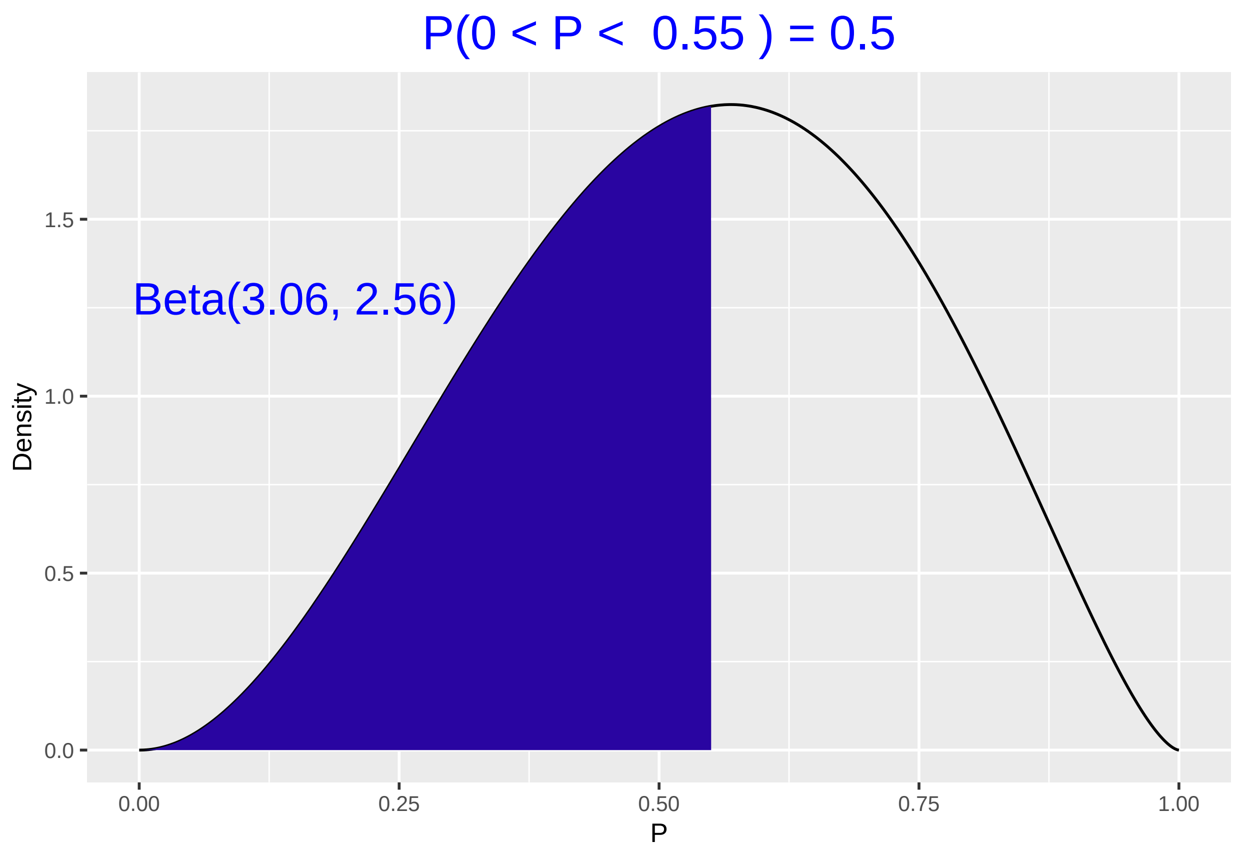


Use `beta.select` to choose a Beta curve

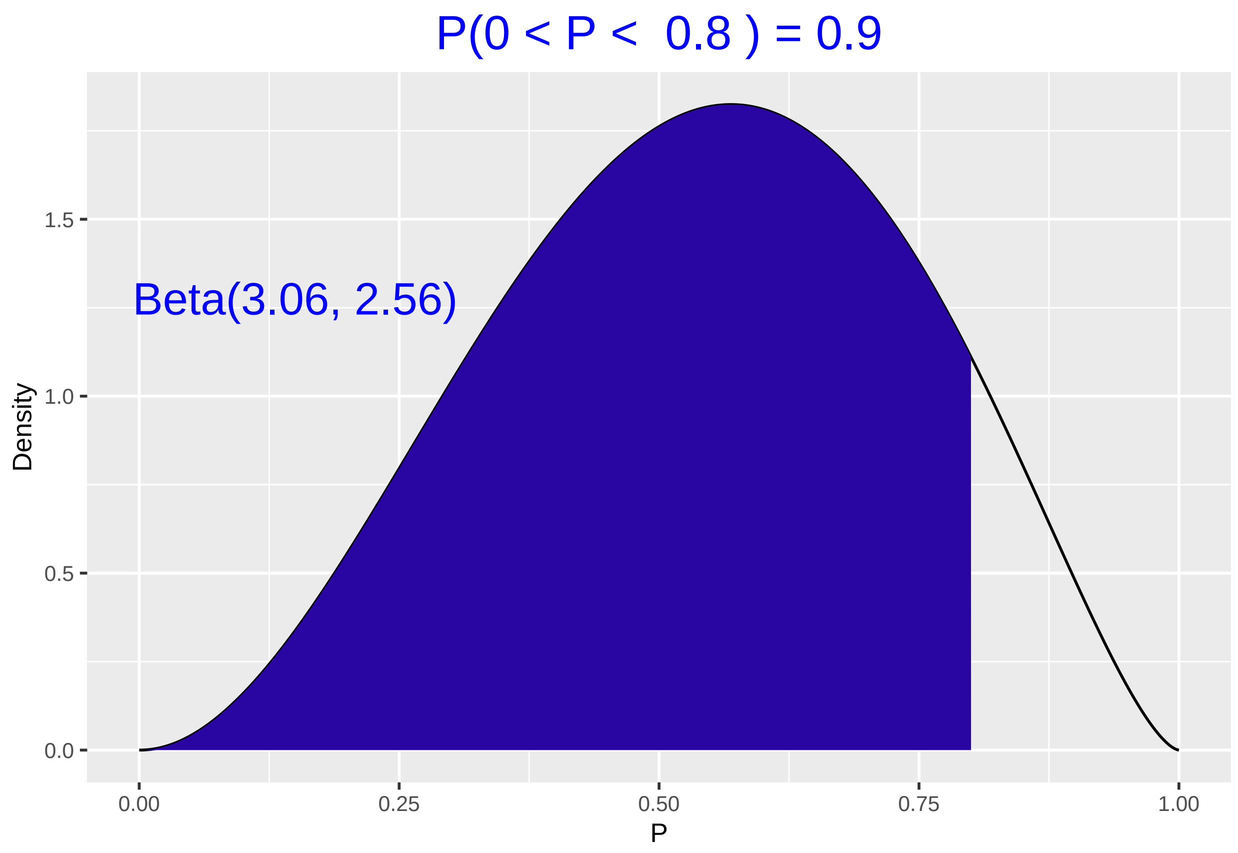
```
beta.select(list(x = 0.55, p = 0.5),
            list(x = 0.80, p = 0.9))
```

```
## [1] 3.06 2.56
```

```
# we can verify that Beta(3.06, 2.56) hat p_50= 0.55
# and p_90 = 0.8

beta_quantile(0.5, c(3.06, 2.56), Color = crcblue) +
  theme(text=element_text(size=10))
```



```
beta_quantile(0.9, c(3.06, 2.56), Color = crcblue) +
  theme(text=element_text(size=10))
```



Updating the Beta prior

Use R/RStudio to compute and plot the posterior

Some derivations:

$p \sim Beta(a,b) \implies \pi(p) = \frac{1}{B(a,b)} p^{a-1}(1-p)^{b-1}$

$y|p \sim Bin(n,p) \implies L(p) = \binom{n}{y} p^y (1-p)^{n-y}$

Want to show: $p|y \sim Beta(a+y, b+n-y)$

$$\begin{aligned}\pi(p,y) &= \frac{\pi(p)L(p)}{\int \pi(\tilde{p})L(\tilde{p})d\tilde{p}} \\ &= \frac{\frac{1}{B(a,b)} p^{a-1}(1-p)^{b-1}\binom{n}{y}p^y(1-p)^{n-y}}{\int \frac{1}{B(a,b)}\tilde{p}^{a-1}(1-\tilde{p})^{b-1}\binom{n}{y}\tilde{p}^y(1-\tilde{p})^{n-y}d\tilde{p}} \\ &= \frac{p^{a+y-1}(1-p)^{b+n-y-1}}{\int \tilde{p}^{a+y-1}(1-\tilde{p})^{b+n-y-1}d\tilde{p}} \\ &= \frac{p^{a+y-1}(1-p)^{b+n-y-1}}{B(a+y, b+n-y)}\end{aligned}$$

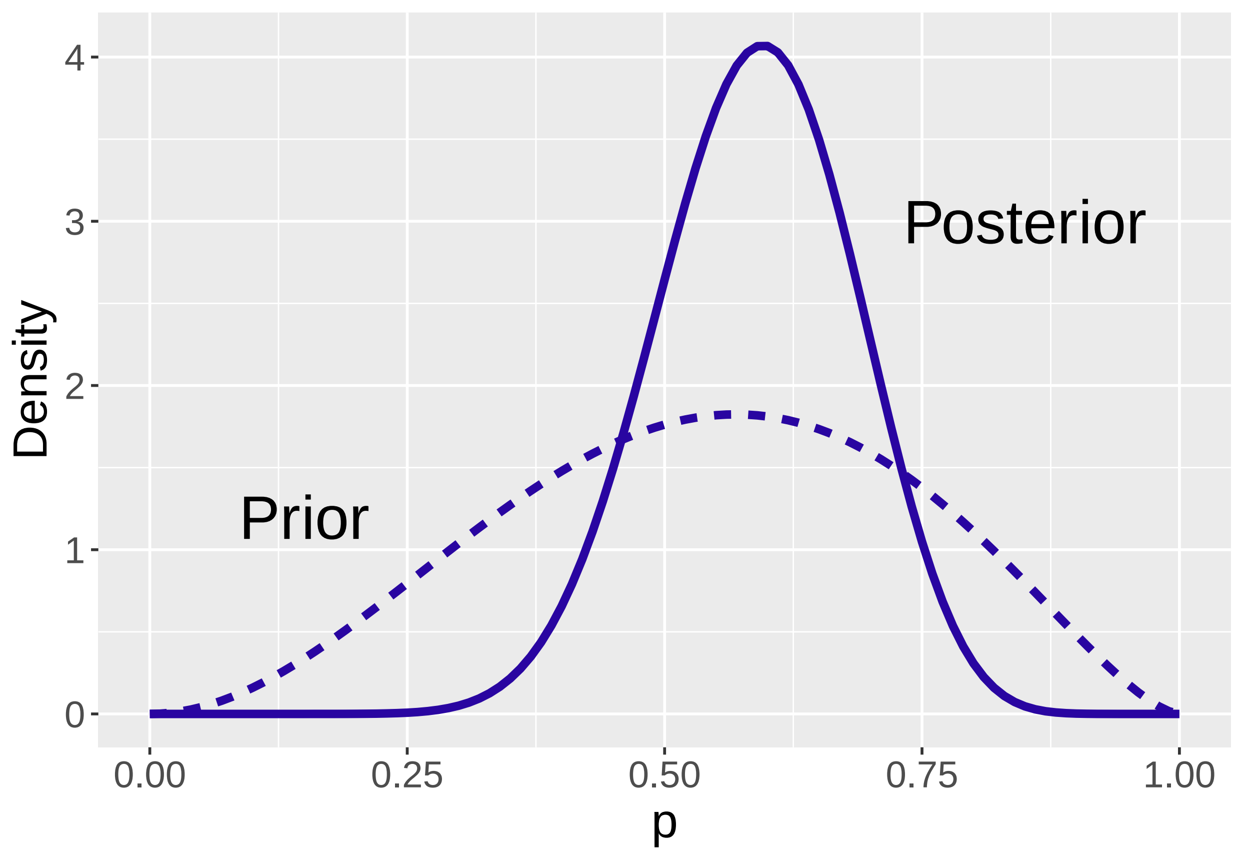
The last equation comes about by the fact that:

$\int \pi(p)dp = 1 \implies \int \frac{1}{B(a+y,b+n-y)}\tilde{p}^{a+y-1}(1-\tilde{p}^{b+n+y-1})d\tilde{p} = 1$

```
ab <- c(3.06, 2.56)
yny <- c(12, 8)
(ab_new <- ab + yny)
```

```
## [1] 15.06 10.56
```

```
ggplot(data.frame(x=c(0, 1)), aes(x)) +
  stat_function(fun=dbeta, geom="line",
                aes(linetype="solid"),
                size=1.5, color = crcblue,
                args=list(shape1=ab[1],
                          shape2=ab[2])) +
  stat_function(fun=dbeta, geom="line",
                aes(linetype="dashed"),
                size=1.5, color = crcblue,
                args=list(shape1=ab_new[1],
                          shape2=ab_new[2])) +
  xlab("p") + ylab("Density") +
  theme(legend.position = "none") +
  annotate(geom = "text", x = .15, y = 1.2,
           size = 8, label = "Prior") +
  annotate(geom = "text", x = .85, y = 3,
           size = 8, label = "Posterior") +
  increasefont()
```



Exercise 1:

Compare prior mean 0.544 and posterior mean 0.588. (Recall that sample mean is $\hat{p} = \frac{y}{n} = 12/20 = 0.6$)

Note that the posterior is a combination of the prior and the data. In fact the posterior is the weighted average of the prior and the data.

This is shown by the fact that the posterior mean (=0.588) is in the middle of the prior mean (=0.544) and the sample mean (=0.6)

If you have strong prior, the posterior will be more heavily influenced by it, o.w., it will be driven more by data.

Please check page 59/90 (or 32/50) of the slides, let's consider two cases of the prior:

1. a:b=1:4
2. a:b=10:40

Even though the ratio is the same, but they carry different amount of information. The larger the prior, the more information you have in the prior, and thus the posterior will be more heavily influenced by the prior in the second case.

Exercise 2:

Compare the spreads/volatility of the two curves. How can we think about the "sharpened belief" once we did the updating?

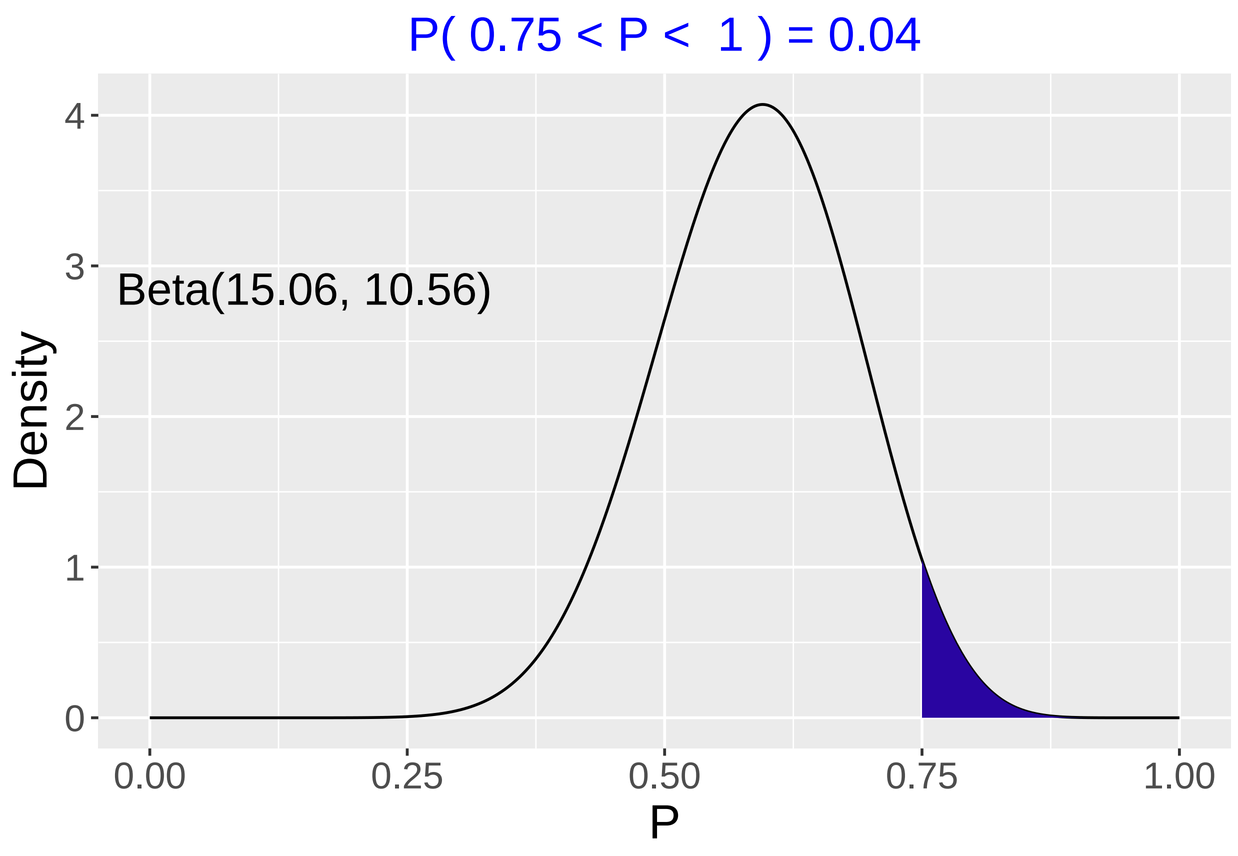
Clearly, the spread on the x-axis of the posterior becomes much smaller, which means that we have less uncertainty about the parameter

Note that both prior and posterior are still densities, so the area under these two curves equals one respectively.

Bayesian inference with continuous priors

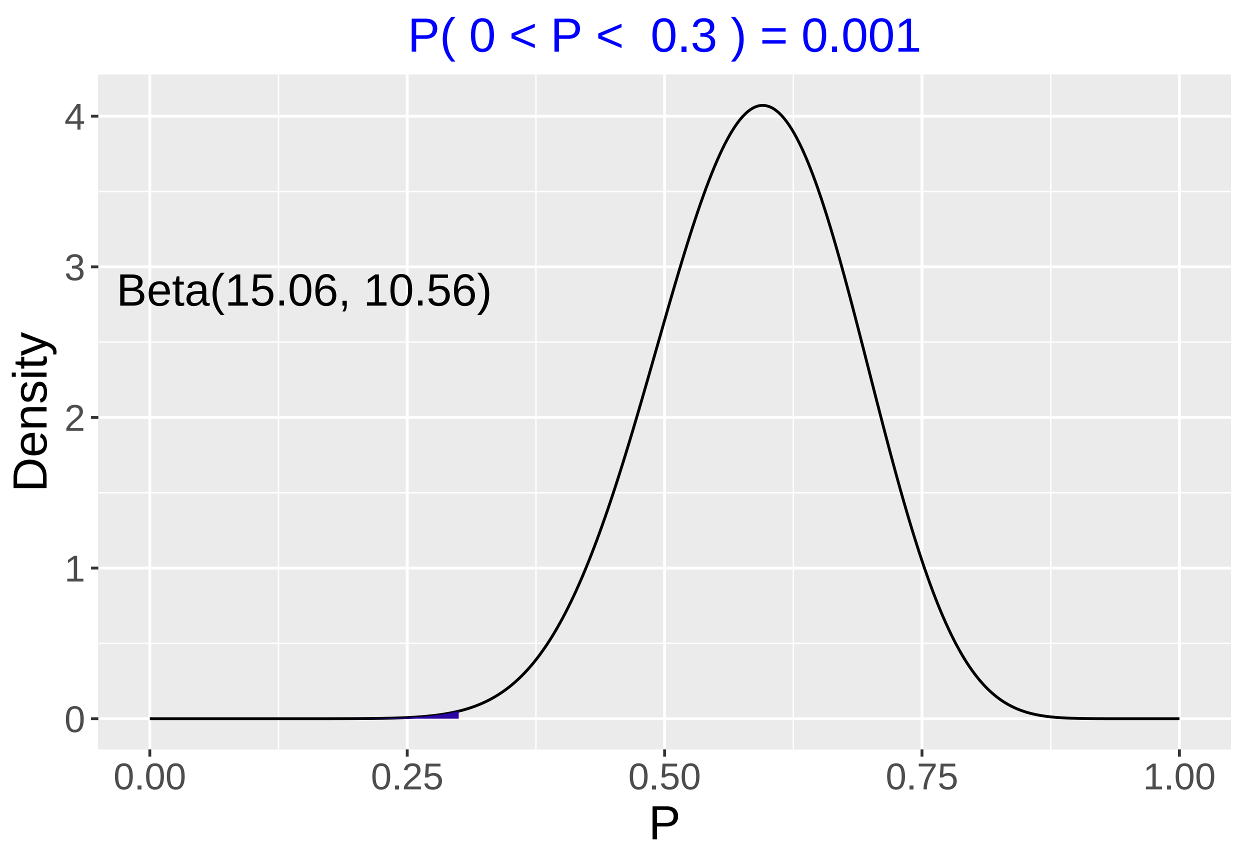
Bayesian hypothesis testing

```
# lo = low, hi = high
beta_area(lo = 0.75, hi = 1.0, shape_par = c(15.06, 10.56),
          Color = crcblue) +
  theme(text=element_text(size=18))
```



Posterior probability is only 4%, reject the claim.

```
# Exercise: what about a claim "at most 30% of the customers
# prefer Friday" ?

beta_area(lo = 0, hi = 0.3, shape_par = c(15.06, 10.56),
          Color = crcblue) +
  theme(text=element_text(size=18))
```



When the posterior distribution is known. . .

- Exact solution: use the `beta_area()` in the `ProbBayes` package and/or the `pbeta()` R function

```
beta_area(lo = 0.75, hi = 1.0, shape_par = c(15.06, 10.56))
```

```
pbeta(1, 15.06, 10.56) - pbeta(0.75, 15.06, 10.56)
```

```
## [1] 0.03973022
```

- Approximation through Monte Carlo simulation using the `rbeta()` R function

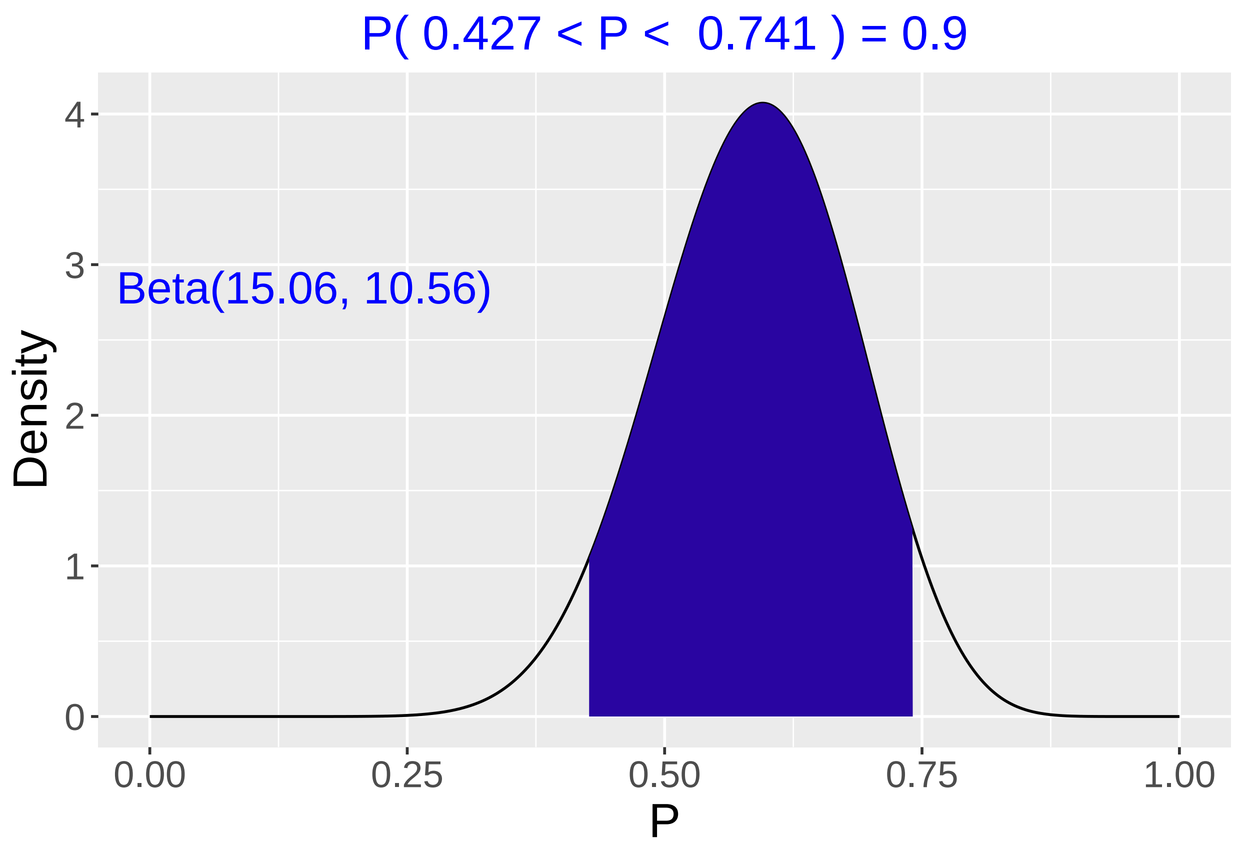
```
S <- 1000 # number of simulation draws
BetaSamples <- rbeta(S, 15.06, 10.56)
sum(BetaSamples >= 0.75)/S
```

```
## [1] 0.042
```

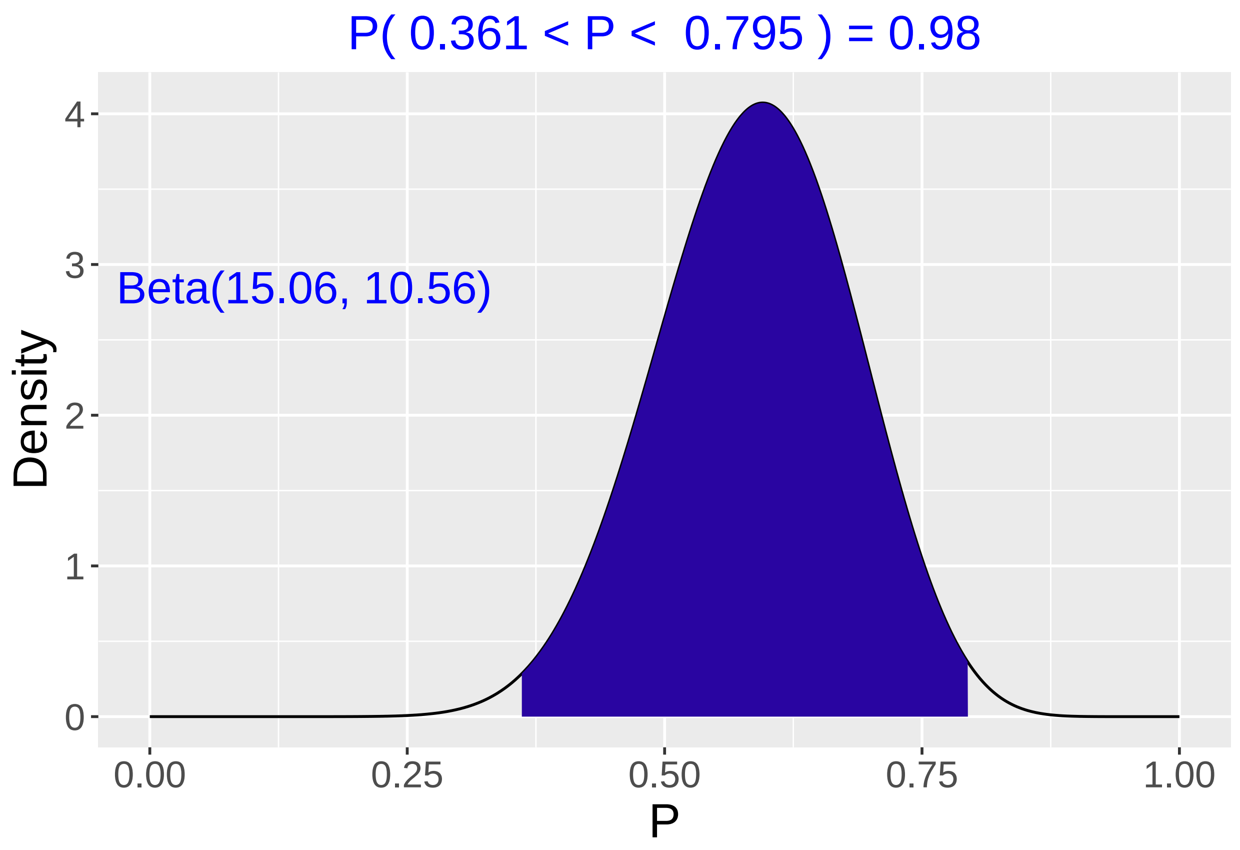
```r
# sum(BetaSamples >= 0.75) tells us how many random draws are
# larger than 0.75
```

Bayesian credible intervals

```r
beta_interval(0.9, c(15.06, 10.56), Color = crcblue) +
  theme(text=element_text(size=18))
```



```r
# Exercise: construct a middle 98% credible interval for p

beta_interval(0.98, c(15.06, 10.56), Color = crcblue) +
  theme(text=element_text(size=18))
```



When the posterior distribution is known. . . .

- Exact solution: use the `beta_interval()` function in the `ProbBayes` package (only the middle 90%) and or the `qbeta()` R function (not necessarily the middle 90%)

```
# middle 90% indeed
c(qbeta(0.05, 15.06, 10.56), qbeta(0.95, 15.06, 10.56))
```

```
## [1] 0.4266788 0.7410141
```

```
# 90%, but not in the middle
c(qbeta(0, 15.06, 10.56), qbeta(0.9, 15.06, 10.56))
```

```
## [1] 0.0000000 0.7099912
```

- Approximation through Monte Carlo simulation using the `rbeta()` R function (not necessarily the middle 90%)

```
# 90% in the middle indeed
S <- 1000; BetaSamples <- rbeta(S, 15.06, 10.56)
quantile(BetaSamples, c(0.05, 0.95))
```

```
##        5%       95%
## 0.4274788 0.7540609
```

```
# 90%, but not in the middle
S <- 1000; BetaSamples <- rbeta(S, 15.06, 10.56)
quantile(BetaSamples, c(0., 0.9))
```

```
##        0%       90%
## 0.2780388 0.7067309
```

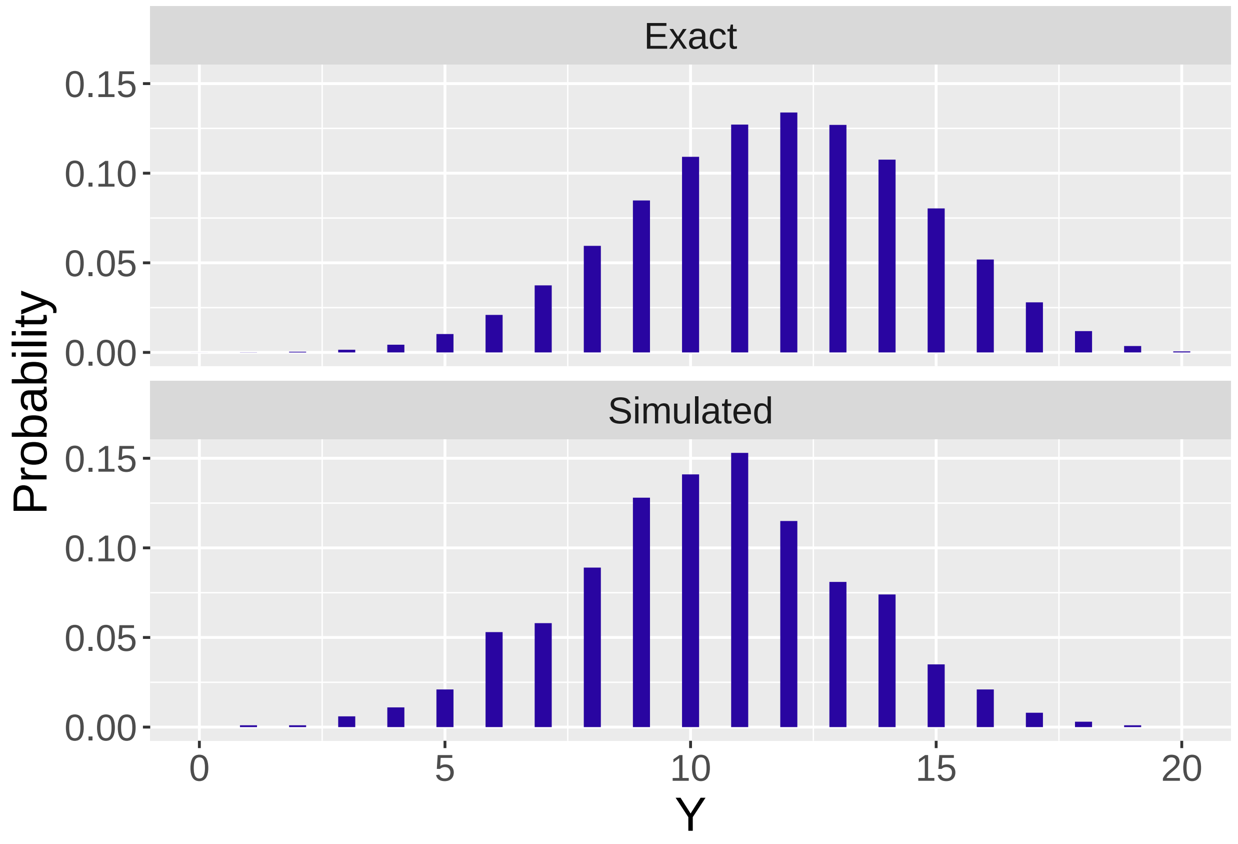
Use R/RStudio to make Bayesian predictions

Bayesian prediction is posterior predictive.

```
S <- 1000
a <- 3.06; b <- 2.56
n <- 20; y <- 12
m <- 20
pred_p_sim <- rbeta(S, a + y, b + n - y) # sample p from posterior
pred_y_sim <- rbinom(S, m, pred_p_sim) # sample new y from p
sum(pred_y_sim >=5 & pred_y_sim <= 15)/S
```

```
## [1] 0.891
```

```
a <- 3.06; b <- 2.56
n <- 20; y <- 12
prob <- pbetap(c(a + y, b + n - y), 20, 0:20)
T1 <- data.frame(Y = 0:10,
                 Probability = round(prob[1:11], 3))
T2 <- data.frame(Y = 11:20,
                 Probability = round(prob[12:21], 3))
T2 <- rbind(T2, data.frame(Y = 21, Probability = 999))

set.seed(123)
S <- 1000
pred_p_sim <- rbeta(S, a + y, a + b + n - y)
pred_y_sim <- rbinom(S, n, pred_p_sim)

data.frame(Y = pred_y_sim) %>%
  group_by(Y) %>% summarize(N = n()) %>%
  mutate(Probability = N / sum(N),
         Type = "Simulated")  %>%
  select(Type, Y, Probability) -> S1

S2 <- data.frame(Type = "Exact",
                 Y = 0:20,
                 Probability = prob)

S <- rbind(S1, S2)
ggplot(S, aes(Y, Probability)) +
  geom_segment(aes(xend = Y, yend = 0),
               size = 3,
               lineend = "butt",
               color = crcblue) +
  facet_wrap(~ Type, ncol=1) +
  theme(text=element_text(size=18))
```



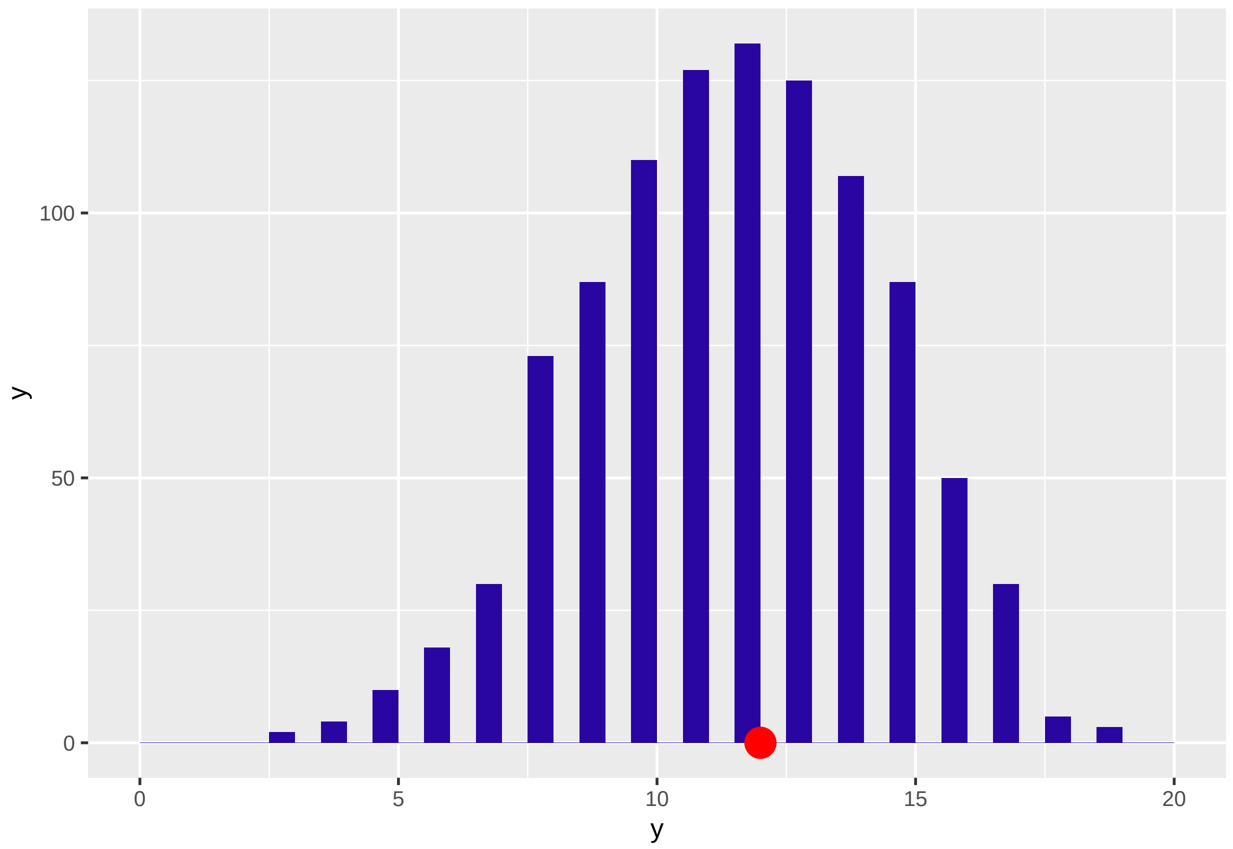
Use R/Rstudio to perform posterior predictive checking

```
S <- 1000
a <- 3.06; b <- 2.56
n <- 20; y <- 12
newy = as.data.frame(rep(NA, S))
names(newy) = c("y")

set.seed(123)
```

```r
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
}
```

```r
ggplot(data=newy, aes(newy$y)) +
  geom_histogram(breaks=seq(0, 20, by=0.5), fill = crcblue) +
  annotate("point", x = 12, y = 0, colour = "red", size = 5) +
  xlab("y") + theme(text=element_text(size=10))
```

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



```r
# If the red dot is in the extreme of the histogram
# then your prediction may not perform well
```

```r
sum(newy > y)/S
```

```
## [1] 0.407
```

```
1 - sum(newy > y)/S
```

```
## [1] 0.593
```

Recap