HW3

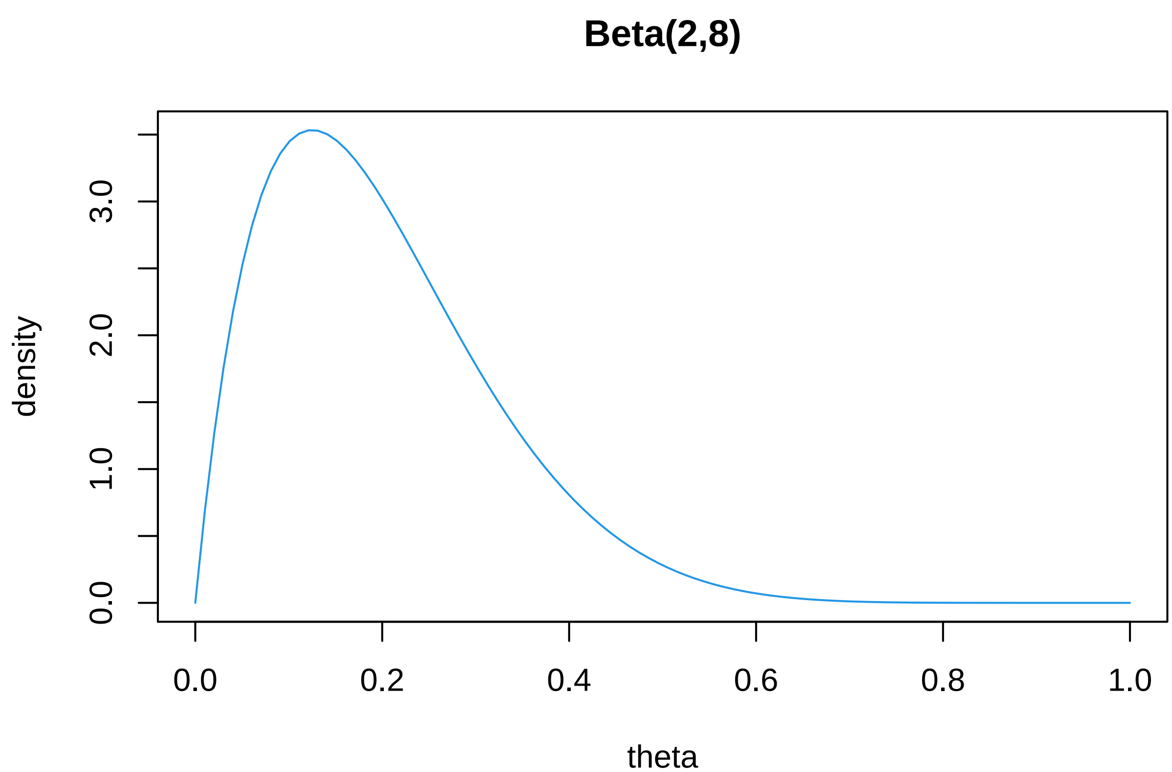
```r
library(bayestestR)
```

```
## Note: The default CI width (currently `ci=0.89`) might change in future versions (see https://github
```

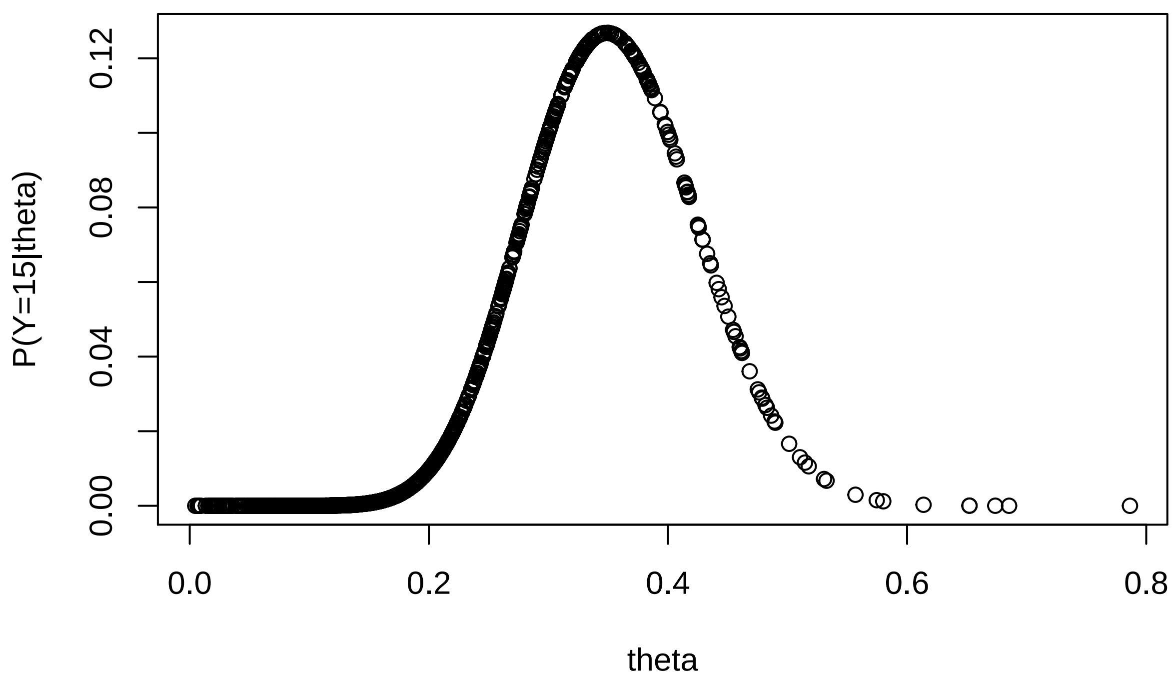
Hoff 3.4

Part A

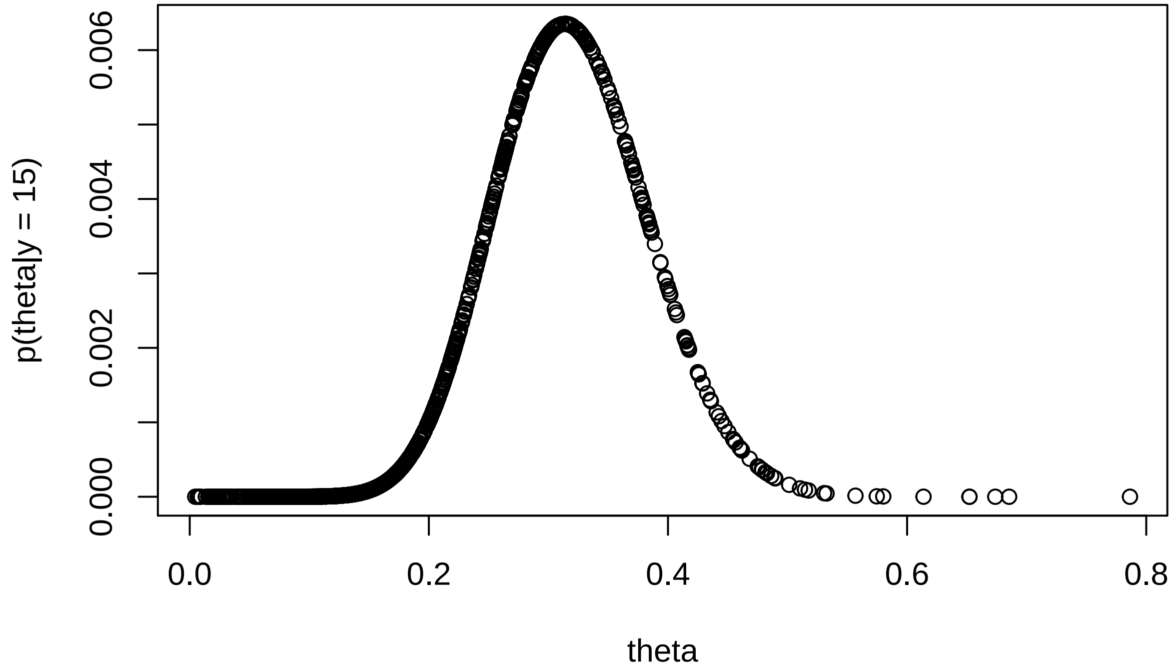
```r
#sample theta from beta(2,8) 1000 times
theta <- rbeta(1000, 2, 8, ncp = 0)

# draw beta dist
p = seq(0,1, length=100)
plot(p, dbeta(p, 2, 8), main = "Beta(2,8)", xlab = "theta", ylab="density", type ="l", col=4)
```



```r
#create empty probability vector
py <- rep(0,1000)

#calc probability for various y|theta
for (i in 1:1000)
{
  py[i] = factorial(43)* (theta[i])^15 * (1-theta[i])^28 /
    (factorial(15) * factorial(28))
}

plot(x = theta, y = py, ylab = "P(Y=15|theta)")
```




```r
#empty p vectors
p_theta <- rep(0,1000)
#calc p of theta|y
for (i in 1:1000)
{
  p_theta[i] = factorial(43) * (theta[i])^15 * (1-theta[i])^28 *
    factorial(9) *
    theta[i] * (1-theta[i])^7 / (factorial(7) * sum(py) *
                                  factorial(15) * factorial(28))
}

plot(x = theta, y = p_theta, ylab = "p(theta|y = 15)")
```




```r
#calc values
mean <- sum(p_theta*theta)
mode <- theta[match(max(p_theta), p_theta)]
var <- sum((theta - mean)^2 * p_theta)
sd <- sqrt(var)
```

Mean

```r
mean
```

```
## [1] 0.4779771
```

Mode

```r
mode
```

```
## [1] 0.3135182
```

Variance

```r
var
```

```
## [1] 0.06050369
```

Standard Deviations

```r
sd
```

```
## [1] 0.245975
```

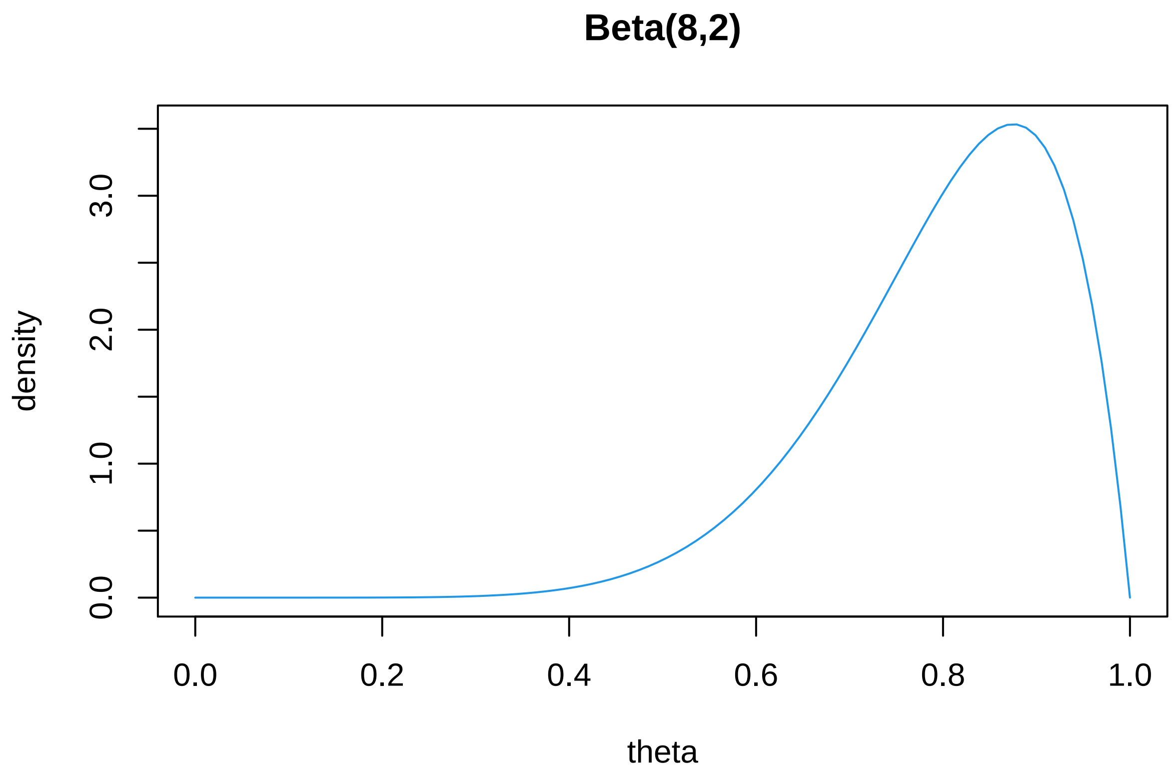
95% Quantile-based Confidence Interval

```r
ci(theta, method = "ETI", ci = .95)
```

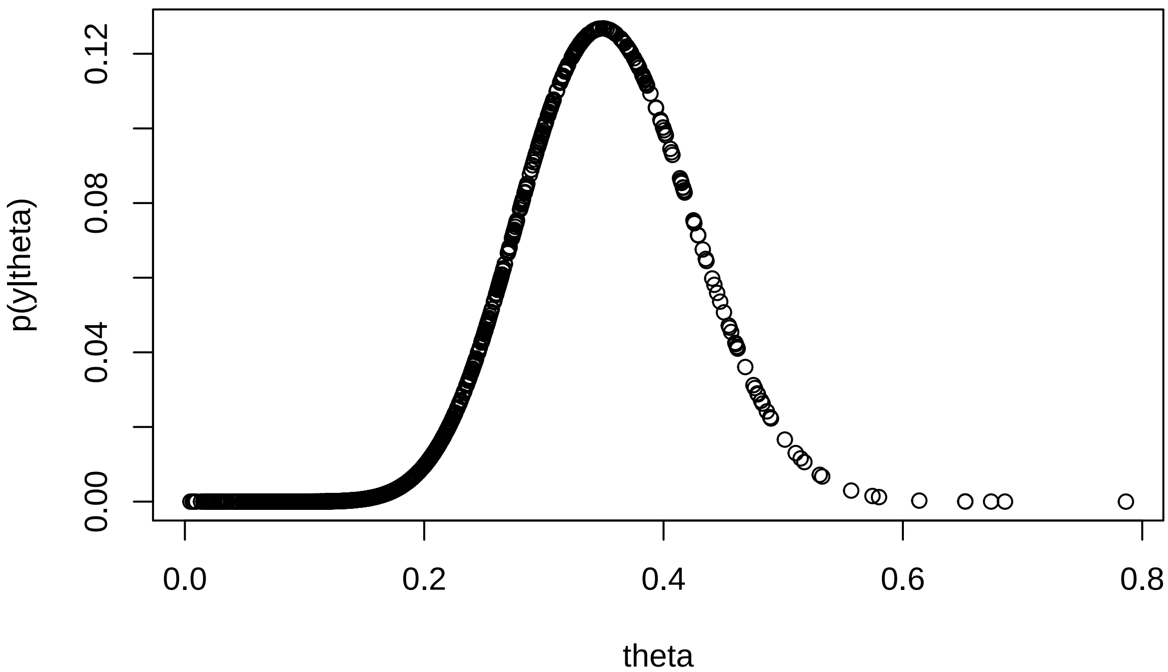
```
## # Equal-Tailed Interval
##
## 95% ETI
## ------------
## [0.03, 0.46]
```

Part B

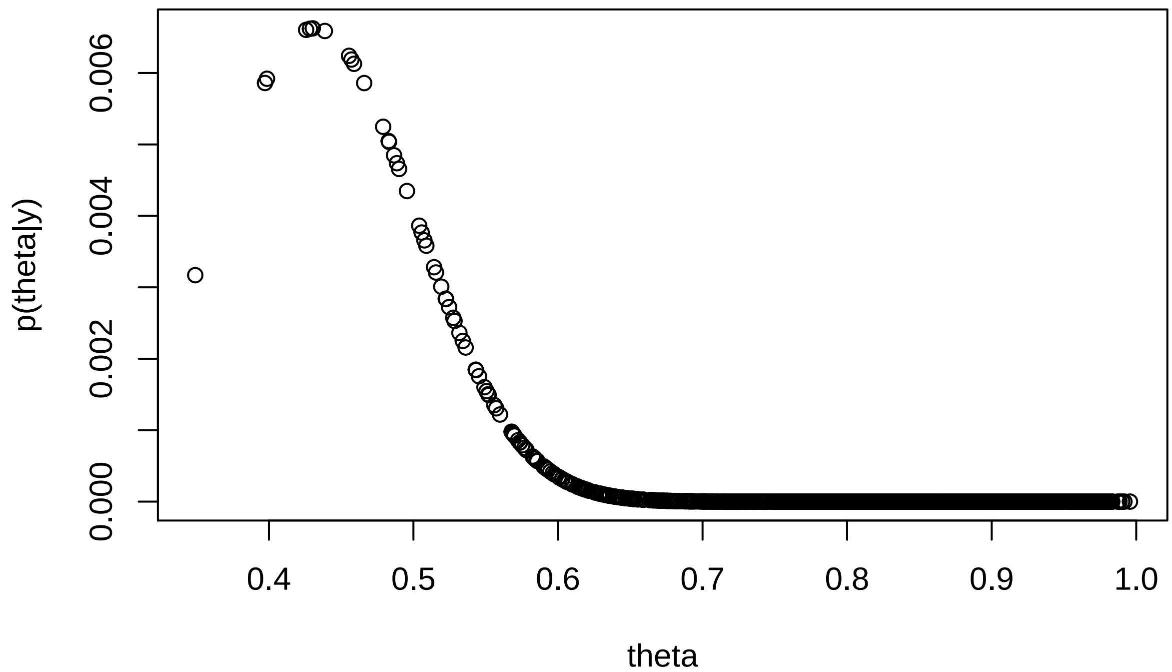
```r
#sample theta from beta(8,2)
theta2 <- rbeta(1000, 8, 2, ncp = 0)
#plot
p2 = seq(0,1, length=100)
plot(p2, dbeta(p, 8, 2), main = "Beta(8,2)", xlab = "theta",
     ylab="density", type ="l", col=4)
```




```
py2 <- rep(0,1000)
#calc probs
for (i in 1:1000)
{
  py2[i] = factorial(43)* (theta2[i])^15 * (1-theta2[i])^28 /
    (factorial(15) * factorial(28))
}

plot(x = theta, y = py, ylab = "p(y|theta)")
```




```r
#create empty p vector
p_theta2 <- rep(0,1000)
#calc p(theta|y)
for (i in 1:1000)
{
  p_theta2[i] = factorial(43) * (theta2[i])^15 * (1-theta2[i])^28 *
    factorial(9) *
    theta2[i]^7 * (1-theta2[i]) / (factorial(7) * sum(py2) *
                                   factorial(15) * factorial(28))
}

plot(x = theta2, y = p_theta2, ylab = "p(theta|y)", xlab = "theta")
```




```
mean2 <- sum(p_theta2*theta2)
mode2 <- theta2[match(max(p_theta2), p_theta2)]
var2 <- sum((theta2 - mean2)^2 * p_theta2)
sd2 <- sqrt(var2)
```

Mean

```
mean2
```

```
## [1] 0.09283387
```

Mode

```
mode2
```

```
## [1] 0.4303708
```

Variance

```r
var2
```

```
## [1] 0.03109011
```

Standard Deviations

```r
sd2
```

```
## [1] 0.1763239
```

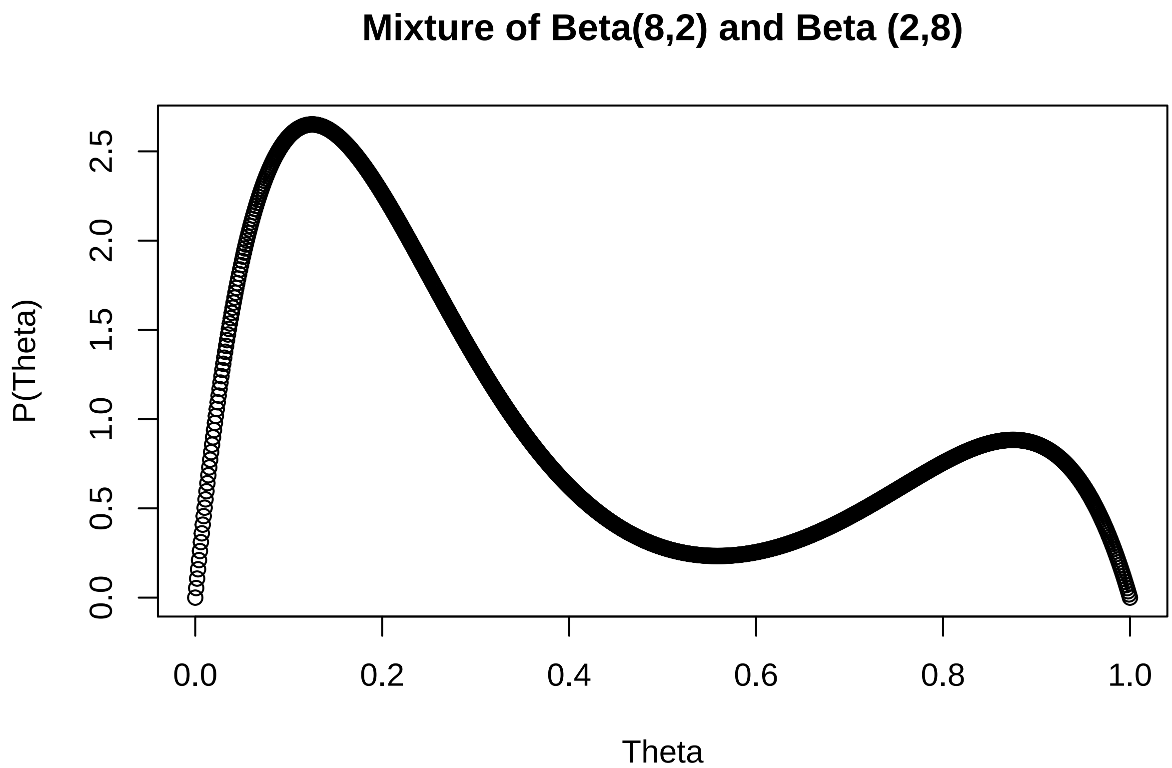
95% Quantile-based Confidence Interval

```r
ci(theta2, method = "ETI", ci = .95)
```

```
## # Equal-Tailed Interval
##
## 95% ETI
## ------------
## [0.52, 0.98]
```

Part C

```r
p_partc <- rep(0, 1000)
#generate thetas on a sequential basis, not sampled from any distribution
theta3 <- seq(0,1, length=1000)
for (i in 1:1000)
{
  p_partc[i] = .25 * factorial(9) *
    (3*theta3[i]*(1-theta3[i])^7 + theta3[i]^7 * (1-theta3[i])) /
    factorial(7)
}
plot(x = theta3, y = p_partc, main = "Mixture of Beta(8,2) and Beta (2,8)", ylab = "P(Theta)", xlab = "
```



The prior distribution appears to be a combination of beta(2,8) and beta(8,2), but with a greater weight on beta(2,8).

This distribution seems to suggest that the prior belief is that there is a 75 percent chance that those released from prison were properly rehabilitated and there will be a low recidivism rate. However, there is also a 25 percent chance that incarceration did not effectively reduce the prisoners' propensity for crime and there will be a high recidivism rate.

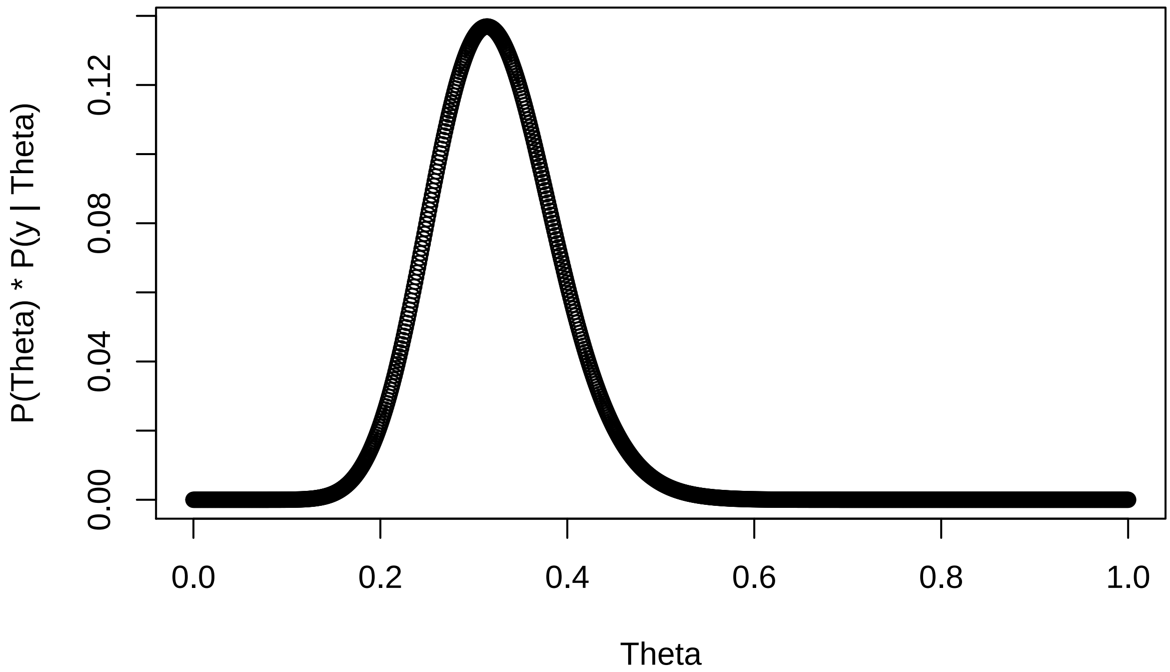
Part D

Part i $P(\Theta) * P(y|\Theta) = \frac{\Gamma(10)[3\theta(1-\theta)^7+\theta^7(1-\theta)]*\binom{43}{15}\theta^{15}(1-\theta)^{28}}{4\Gamma(2)\Gamma(8)}$
$= \binom{43}{15} * 18[3\theta^{16}(1-\theta)^{35} + \theta^{22}(1-\theta)^{29}]$

Part ii The Posterior is a combination of two beta distributions, beta(17,36) and beta(23, 30).

```
p_partd <- rep(0, 1000)

for (i in 1:1000)
{
  p_partd[i] = .25 * factorial(9) *
    (3*theta3[i]^16 * (1-theta3[i])^35 + theta3[i]^22 * (1-theta3[i])^29) * factorial(43) /
    (factorial(7) * factorial(15) * factorial(28))
```

```
}
plot(x = theta3, y = p_partd, ylab = "P(Theta) * P(y | Theta)", xlab = "Theta")
```



Part iii

```
#mode
theta3[match(max(p_partd), p_partd)]
```

```
## [1] 0.3143143
```

The mode of ~.314 is much closer to the mode in Part A (~.313) than the mode from Part B (~.4288).

Part E

~ In the written portion ~

Hoff 3.9

Part A

```
dgalenshore <- function(x, a, theta) {
    # takes numeric vector (elements should be >0) and returns the galenshore
```

```
    # probability density at each point. Parameters a and theta should all be
    # > 0.
    if (any(x <= 0) || any(a <= 0) || any(theta <= 0)) {
        stop("Invalid parameters or input values. Inputs must be >0")
    } else {
        return((2/gamma(a)) * theta^(2 * a) * exp(-theta^2 * x^2))
    }
}

x <- seq(0.01, 3, 0.01)
plot(x, dgalenshore(x, 1, 1), type = "l", ylab = "Galenshore Density(x)", col = "black",
    ylim = c(0, 5), lwd = 2)
lines(x, dgalenshore(x, 1, 2), type = "l", col = "red", lwd = 2, lty = 2)
lines(x, dgalenshore(x, 3, 1), type = "l", col = "blue", lwd = 2, lty = 3)
lines(x, dgalenshore(x, 1, .8), type = "l", col = "green", lwd = 2, lty = 2)
lines(x, dgalenshore(x, 5, 2), type = "l", col = "purple", lwd = 2, lty = 1)
lines(x, dgalenshore(x, 5, 1), type = "l", col = "orange", lwd = 2, lty = 1)
legend("topright", c("Galenshore(1,1)", "Galenshore(1,2)", "Galenshore(3,1)", "Galenshore(1,.8)", "Galen
    col = c("black", "red", "blue", "green", "purple", "orange"), lwd = c(2, 2, 2), lty = c(1, 2, 3, 2,
```