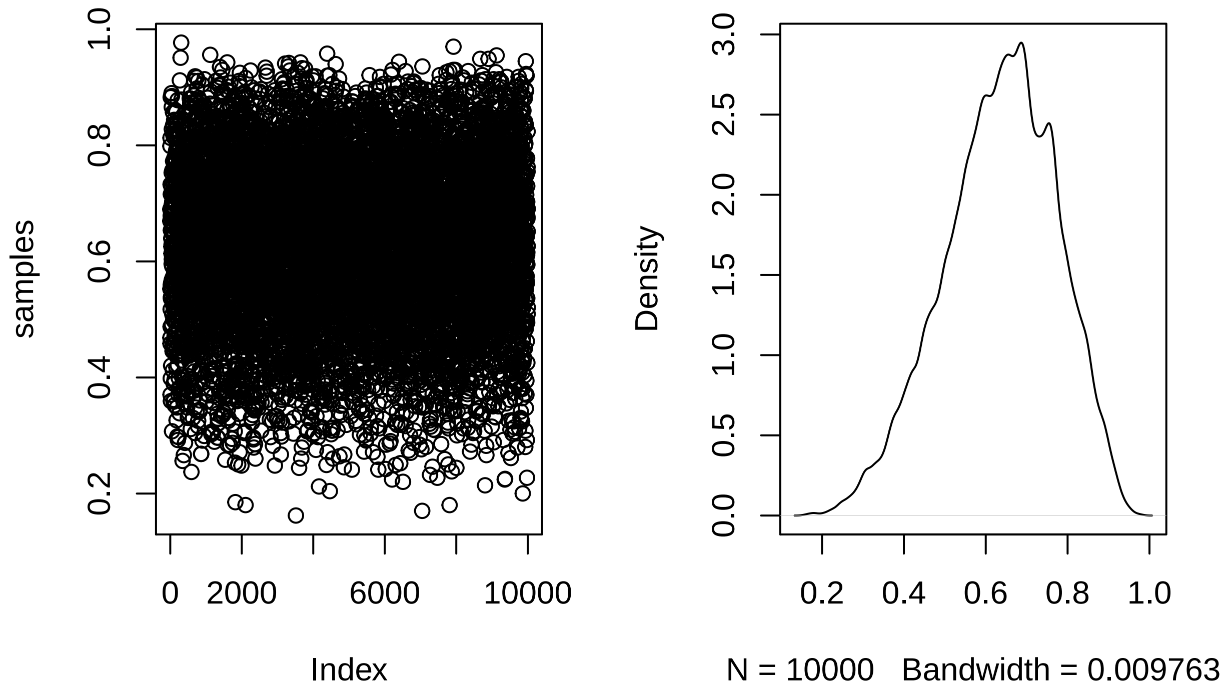
Chapter 3 – Practice

```
p_grid = seq(from=0, to=1, length.out=1000)
prior <- rep(1, 1000)
likelihood <- dbinom(6, size=9, prob=p_grid)
posterior <- likelihood * prior
posterior <- posterior / sum(posterior)
set.seed(100)
samples <- sample(p_grid, prob=posterior, size=1e4, replace=T)
```

```
par(mfrow=c(1, 2))
plot(samples)
plot(density(samples, adjust = 0.5), main="")
```



3E1. How much posterior probability lies below p = 0.2?

```
sum(posterior[p_grid < 0.2])
```

```
## [1] 0.0008560951
```

```
sum(samples < 0.2) / length(samples)
```

```
## [1] 5e-04
```

3E2. How much posterior probability lies above p = 0.8?

```
sum(posterior[p_grid > 0.8])
```

```
## [1] 0.1203449
```

```
sum(samples > 0.8) / length(samples)
```

```
## [1] 0.1117
```

3E3. How much posterior probability lies between p = 0.2 and p = 0.8?

```
sum(posterior[p_grid > 0.2 & p_grid < 0.8])
```

```
## [1] 0.878799
```

```
sum(samples > 0.2 & samples < 0.8) / length(samples)
```

```
## [1] 0.8878
```

3E4. 20% of the posterior probability lies below which value of p?

```
quantile(samples, 0.2)
```

```
##       20%
## 0.5195195
```

3E5. 20% of the posterior probability lies above which value of p?

```
quantile(samples, 0.8)
```

```
##       80%
## 0.7567568
```

3E6. Which values of p contain the narrowest interval equal to 66% of the posterior probability?

```
samples_for_hpdi <- coda::as.mcmc(samples)
# x <- sapply(0.66, function(p) coda::HPDinterval(samples_for_hpdi,  prob = p))
x <- coda::HPDinterval(samples_for_hpdi, prob=0.66)
c(x[1], x[2])
```

```
## [1] 0.5205205 0.7847848
```

```
HPDI(samples, prob=0.66)
```

```
##     |0.66     0.66|
## 0.5205205 0.7847848
```

3E7. Which values of p contain 66% of the posterior probability, assuming equal posterior probability both below and above the interval?

```
low = (1 - 0.66) / 2
up = low + 0.66
interval = c(low, up)
c(interval, interval[2] - interval[1])
```

```
## [1] 0.17 0.83 0.66
```

```
quantile(samples, interval)
```

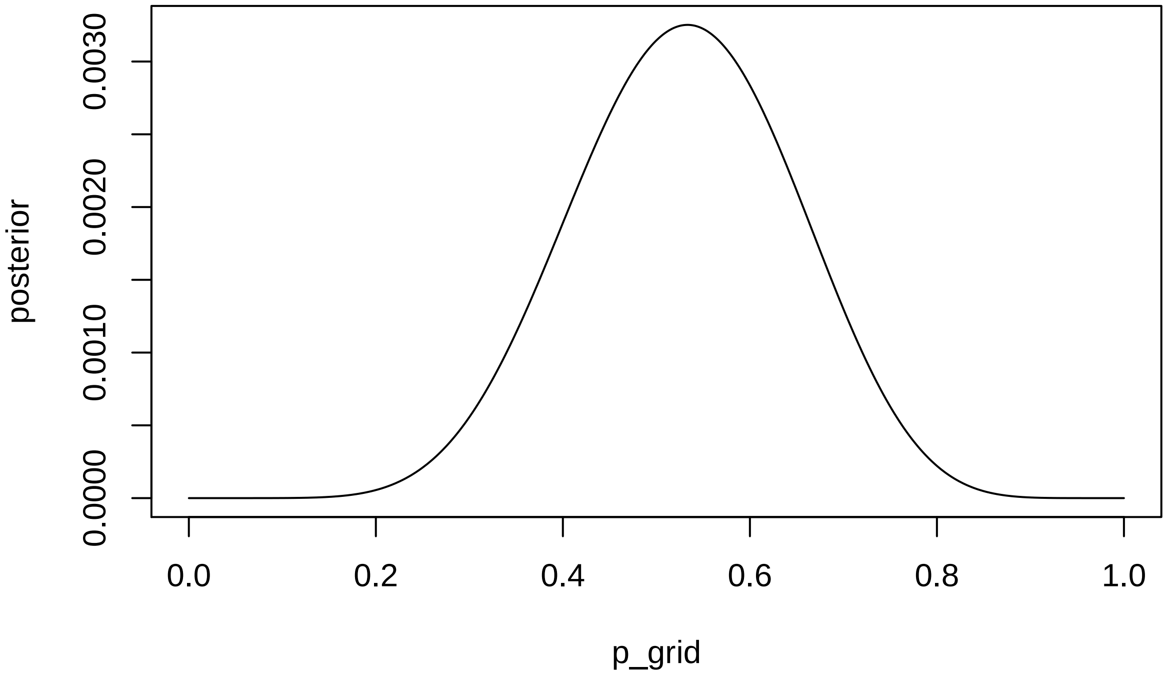
```
##       17%       83%
## 0.5005005 0.7687688
```

If the distribution is not too skewed then the Percentile Interval (PI) will approximately equal to the Highest Posterior Density Interval (HPDI):

- PI: (0.501, 0.769)
- HPDI: (0.521, 0.785)

3M1. Suppose the globe tossing data had turned out to be 8 water in 15 tosses. Construct the posterior distribution, using grid approximation. Use the same flat prior as before.

```
N = 1000
p_grid <- seq(0, 1, length.out = N)
prior <- rep(1, N)
likelihood <- dbinom(8, size=15, prob=p_grid)
unstd_posterior <- likelihood * prior
posterior <- unstd_posterior / sum(unstd_posterior)
plot(posterior ~ p_grid, type="l")
```



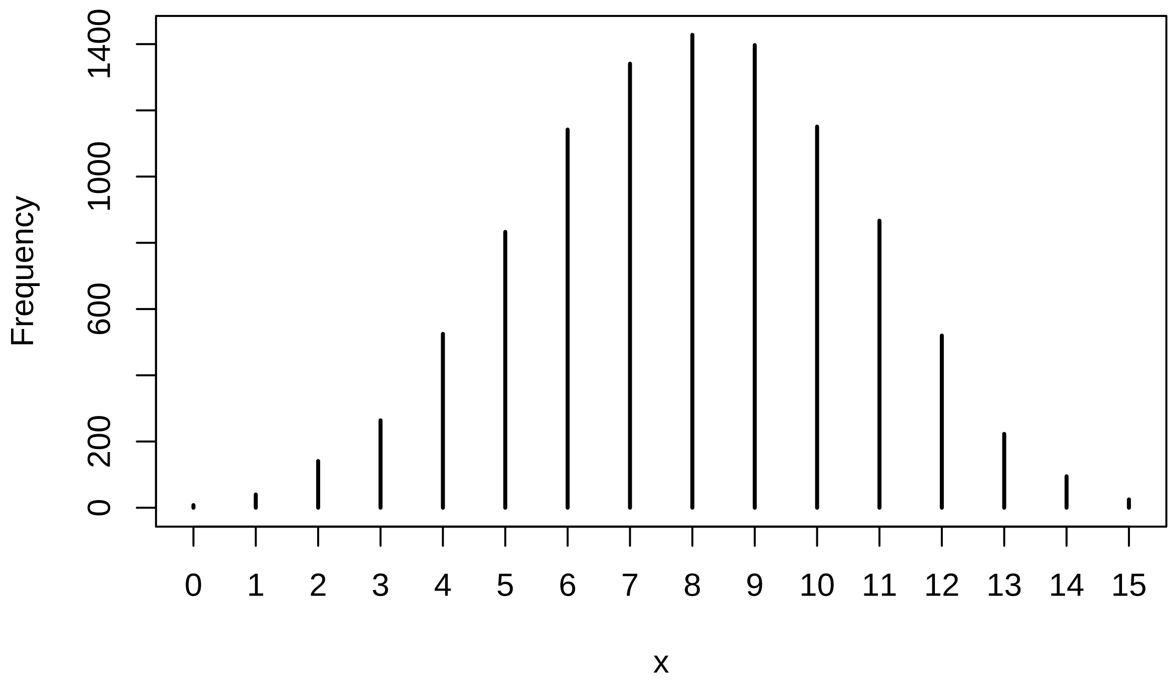
3M2. Draw 10,000 samples from the grid approximation from above. Then use the samples to calculate the 90% HPDI for *p*.

```
samples <- sample(p_grid, size=1e4, replace = T, prob=posterior)
samples_for_hpdi <- coda::as.mcmc(samples)
x <- coda::HPDinterval(samples_for_hpdi, prob=0.90)
c(x[1], x[2])
```

```
## [1] 0.3383383 0.7317317
```

3M3. Construct a posterior predictive check for this model and data. Simulate the distribution of samples, averaging over the posterior uncertainty in *p*. What is the probability of observing 8 water in 15 tosses.

```
w <- rbinom(1e4, size=15, prob=samples)
simplehist(w)
```



Probability of observing 8 water in 15 tosses:

```
sum(w == 8) / length(w)
```

```
## [1] 0.1428
```

```
mean(w == 8)
```

```
## [1] 0.1428
```

3M4. Using the posterior distribution constructed from the new (8/15) data, now calculate the probability of observing 6 water in 9 tosses.

```
w6 <- rbinom(1e4, size=9, prob=samples)
mean(w6 == 6)
```

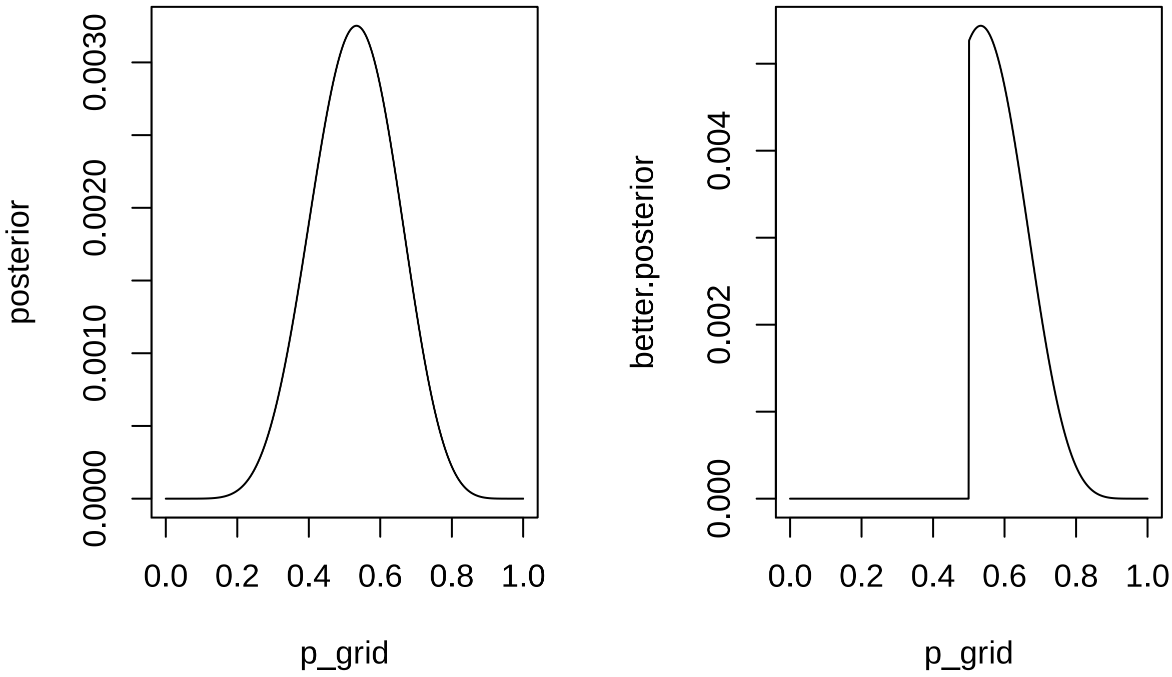
```
## [1] 0.1695
```

```
sum(w6 == 6) / length(w6)
```

```
## [1] 0.1695
```

3M5. Start over at 3M1, but now use a prior that is zero below $p = 0.5$ and a constant above $p = 0.5$. This corresponds to prior information that a majority of the Earth's surface is water. Repeat each problem above and compare inferences. What difference does the better prior make? If it helps, compare inferences (using both priors) to the true value $p = 0.7$.

```
better.prior <- rep(1, N)
better.prior[p_grid < 0.5] = 0
better.unstd_posterior = likelihood * better.prior
better.posterior = better.unstd_posterior / sum(better.unstd_posterior)
par(mfrow=c(1, 2))
plot(posterior ~ p_grid, type="l")
plot(better.posterior ~ p_grid, type="l")
```



Repeat 3M2. Draw 10,000 samples; calculate the 90% HPDI for p:

```
better.samples <- sample(p_grid, size=1e4, replace = T, prob=better.posterior)
better.samples_for_hpdi <- coda::as.mcmc(better.samples)
better.x <- coda::HPDinterval(better.samples_for_hpdi, prob=0.90)
```

Previous posterior:

```
x
```

```
##          lower     upper
## var1 0.3383383 0.7317317
## attr(,"Probability")
## [1] 0.9
```

Better posterior:

```
better.x
```

```
##          lower     upper
## var1 0.5005005 0.7097097
## attr(,"Probability")
## [1] 0.9
```

Better posterior has a much narrower 90% HPDI that's centered around true value of $p = 0.7$.

Repeat 3M3.

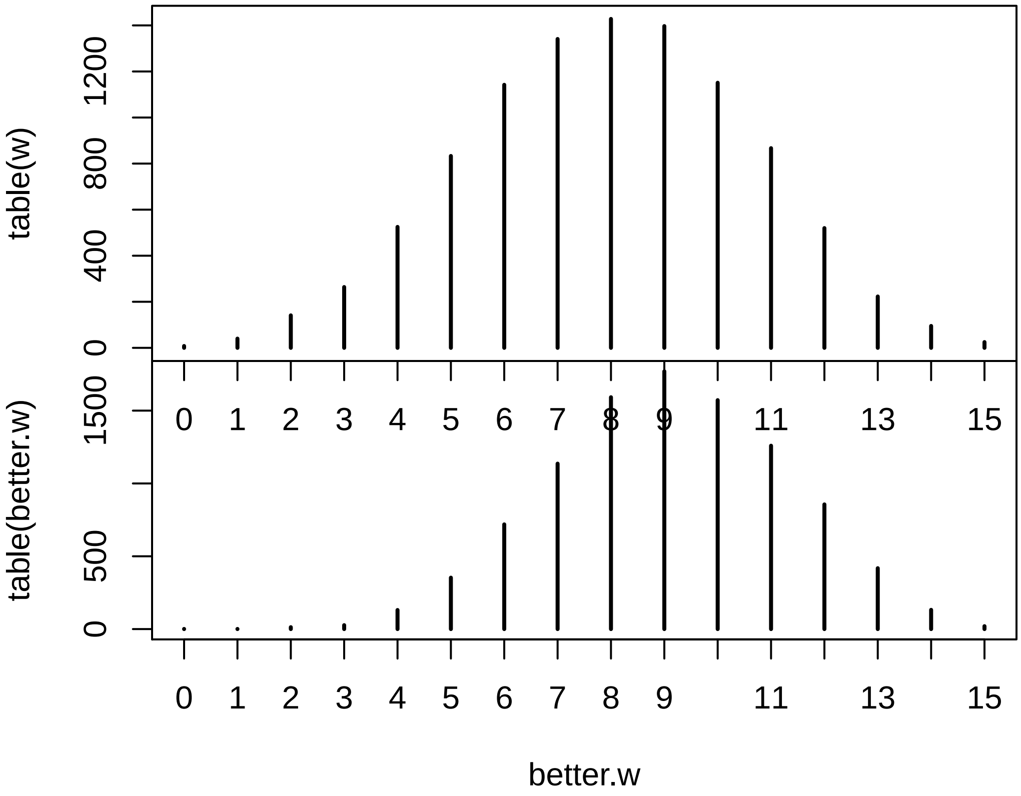
```
better.w <- rbinom(1e4, size=15, prob=better.samples)
layout(rbind(1, 2))
par(mar=c(0,5,2,5))
table(w)
```

```
## w
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14 
##    8   40  141  264  525  833 1142 1341 1428 1397 1151  867  520  223   95 
##   15 
##   25
```

```
plot(table(w))
par(mar=c(4,5,0,5))
better.w[1] <- 0 # to add one zero value to align graphs
table(better.w)
```

```
## better.w
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14 
##    1    1   13   27  131  353  719 1136 1592 1770 1572 1259  856  418  132 
##   15 
##   20
```

```
plot(table(better.w))
```



Probability of observing 8 water in 15 tosses:

```
mean(w == 8)
```

```
## [1] 0.1428
```

```
mean(better.w == 8)
```

```
## [1] 0.1592
```

Repeat 3M4.

```
better.w6 <- rbinom(1e4, size=9, prob=better.samples)
mean(w6 == 6)
```

```
## [1] 0.1695
```

```
mean(better.w6 == 6)
```

```
## [1] 0.2357
```

Hard

```
birth1 <- c(1,0,0,0,1,1,0,1,0,1,0,0,1,1,0,1,1,0,0,0,1,0,0,0,1,0,
0,0,0,1,1,1,0,1,0,1,1,1,0,1,0,1,1,0,1,0,0,1,1,0,1,0,0,0,0,0,0,0,
1,1,0,1,0,0,1,0,0,0,1,0,0,1,1,1,1,0,1,0,1,1,1,1,1,0,0,1,0,1,1,0,
1,0,1,1,1,0,1,1,1,1)
```

```
birth2 <- c(0,1,0,1,0,1,1,1,0,0,1,1,1,1,1,0,0,1,1,1,0,0,1,1,1,0,
1,1,1,0,1,1,1,0,1,0,0,1,1,1,1,0,0,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,0,1,1,0,1,1,0,1,1,1,0,0,0,0,0,0,1,0,0,0,1,1,0,0,1,0,0,1,1,
0,0,0,1,1,1,0,0,0,0)

sum(birth1) + sum(birth2) # Should be 111
```

```
## [1] 111
```

```
length(birth1) + length(birth2)
```

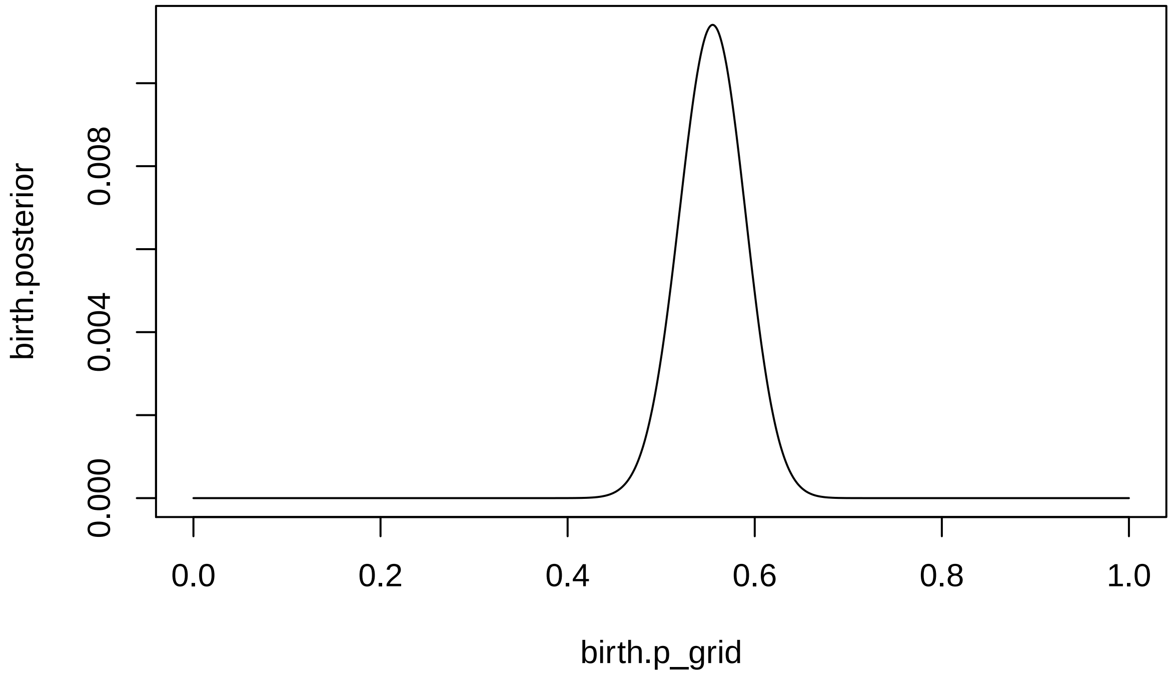
```
## [1] 200
```

3H1. Using grid approximation, compute the posterior distribution for the probability of a birth being a boy. Assume a uniform prior. Which parameter value maximizes the posterior probability?

```
birth.N = 1000
birth.p_grid = seq(0, 1, length.out = N)
birth.all = c(birth1, birth2)
length(birth.all)
```

```
## [1] 200
```

```
birth.prior = rep(1, birth.N)
birth.likelihood = dbinom(sum(birth.all), size=length(birth.all), prob=birth.p_grid)
birth.posterior = birth.likelihood * birth.prior / sum(birth.likelihood * birth.prior)
plot(birth.posterior ~ birth.p_grid, type="l")
```



```
birth.p_grid[which.max(birth.posterior)]
```

```
## [1] 0.5545546
```

3H2. Using the sample function, draw 10,000 random parameter values from the posterior distribution. Use this samples to estimate the 50%, 89%, 97% highest posterior density intervals.

```
birth.samples <- sample(birth.p_grid, size=1e4, replace = T, prob = birth.posterior)
birth.samples_for_hpdi <- coda::as.mcmc(birth.samples)
```

- 50% HPDI:

```
coda::HPDinterval(birth.samples_for_hpdi, prob=0.50)[1:2]
```

```
## [1] 0.5255255 0.5725726
```

- 89% HPDI:

```
coda::HPDinterval(birth.samples_for_hpdi, prob=0.89)[1:2]
```

```
## [1] 0.5015015 0.6116116
```

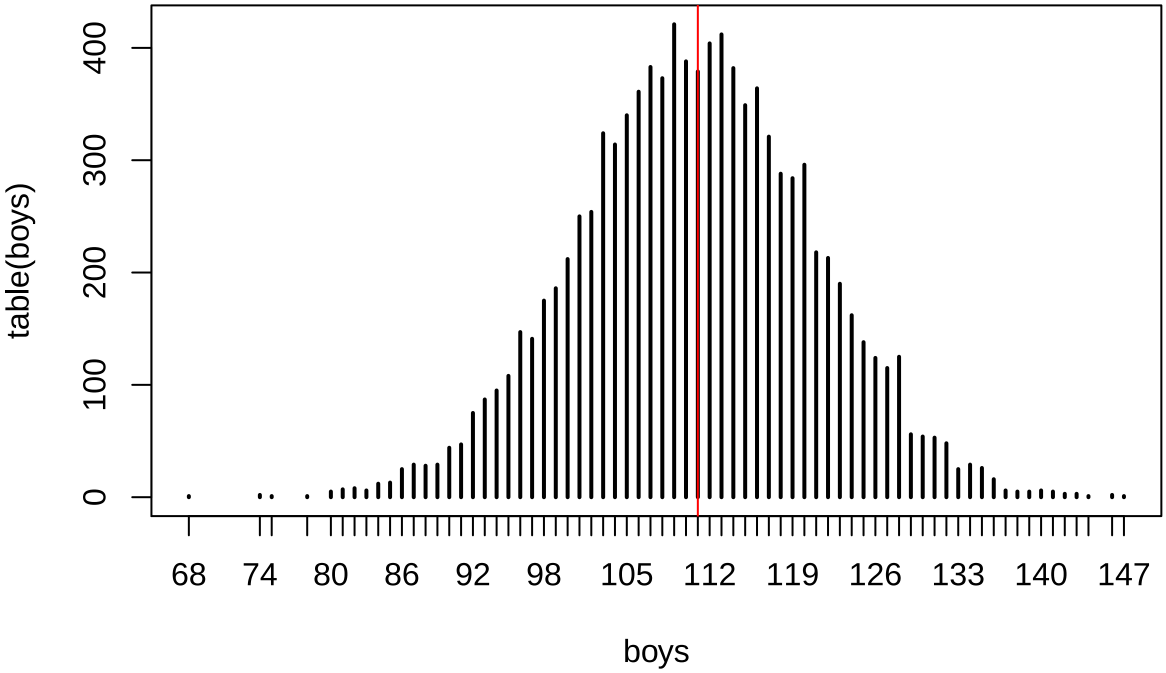
- 97% HPDI:

```
coda::HPDinterval(birth.samples_for_hpdi, prob=0.97)[1:2]
```

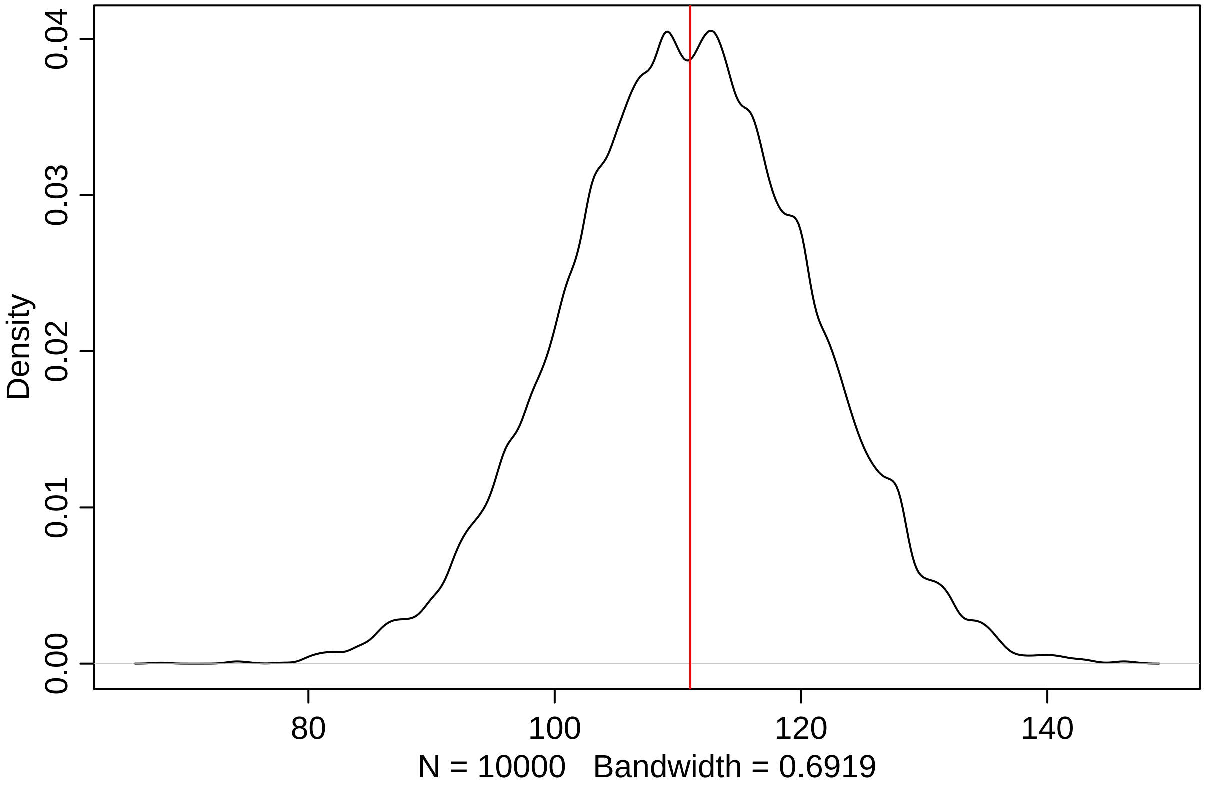
```
## [1] 0.4764765 0.6286286
```

3H3. Use rbinom to simulate 10,000 replicates of 200 births. You should end up with 10,000 numbers, each one a count of boys out of 200. Compare the distribution of predicted numbers of boys to the actual count in the data (111 boys out of 200 births). There are many good ways to visualize the simulations, but the *dens* commmand is probably the easiest way. Does it look like the model fits the data well? That is, does the distribution of predictions include the actual observatin as a central, likely outcome?

```
boys <- rbinom(n=1e4, size=200, prob=birth.samples)
plot(table(boys))
abline(v=sum(birth.all), col="red")
```



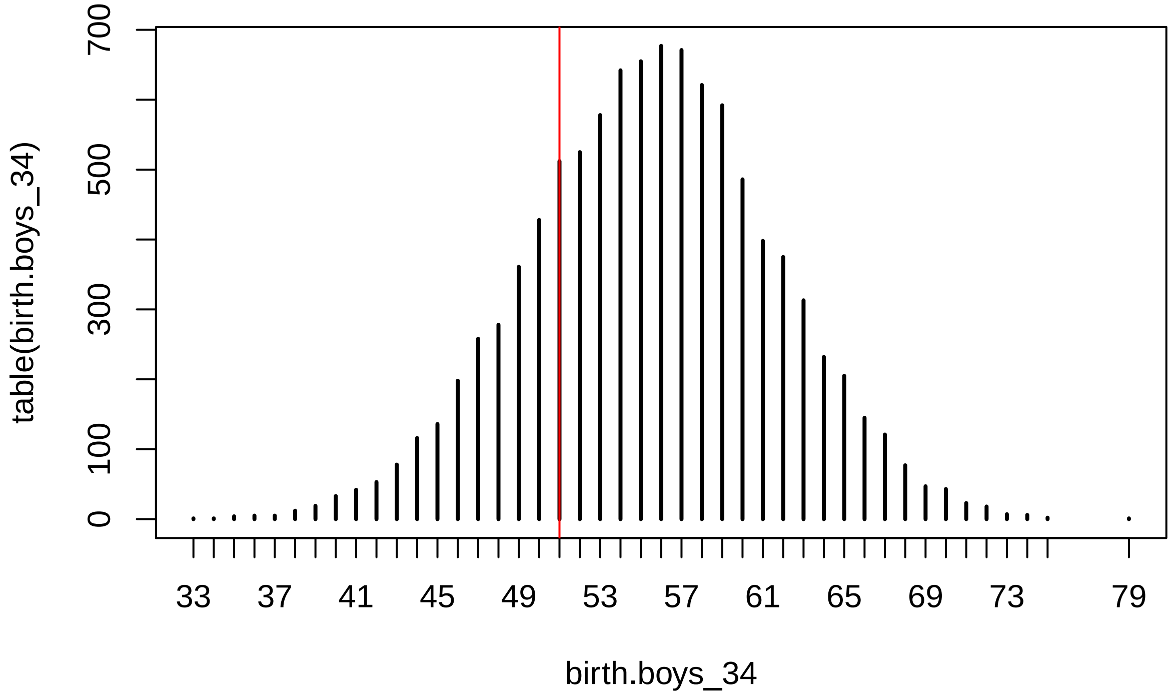
```
dens(boys)
abline(v=sum(birth.all), col="red")
```



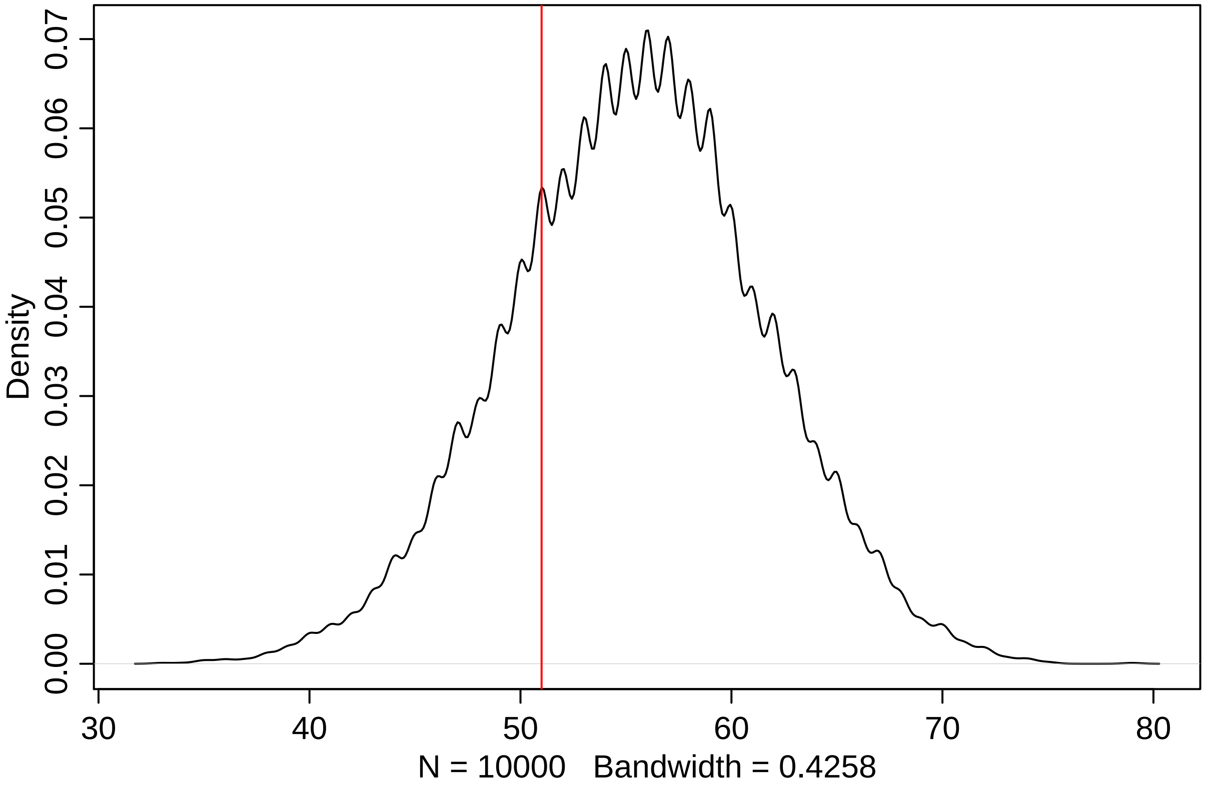
The distributio reflects the actual result, with *boys = 111* being the central, very likely outcome.

3H4. Now compare 10,000 counts of boys from 100 simulated first borns only to the number of boys in *birth1*. How does the model look in this light?

```
birth.boys_34 <- rbinom(1e4, size=100, prob=birth.samples)
plot(table(birth.boys_34))
abline(v=sum(birth1), col="red")
```



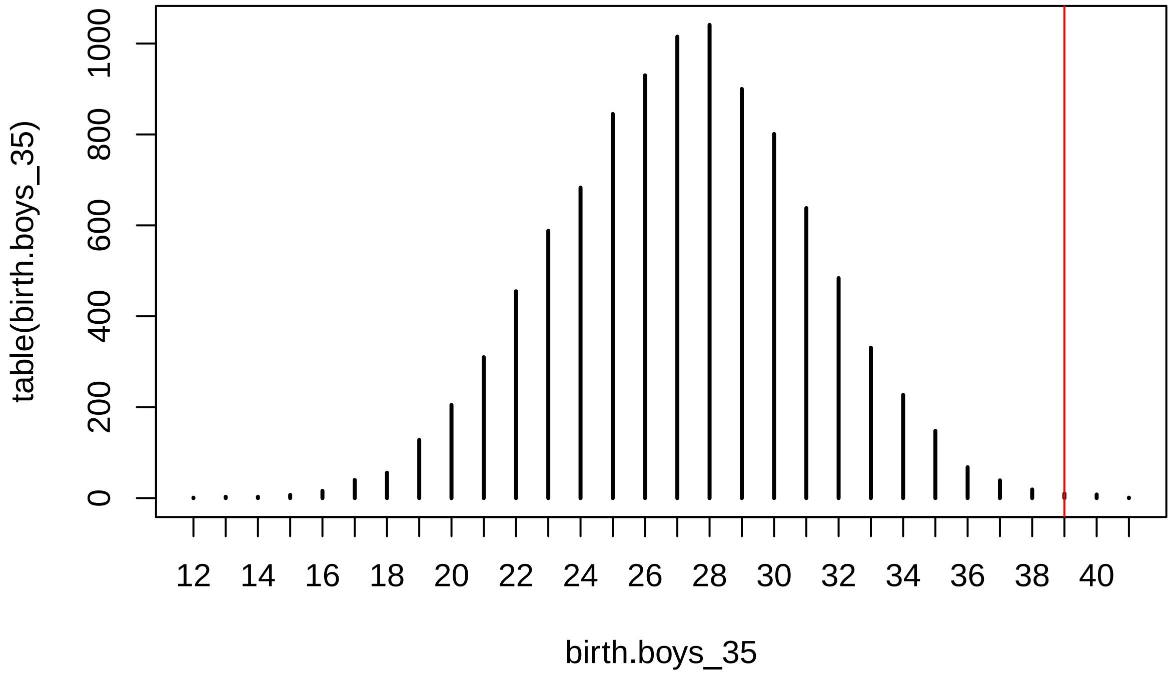
```
dens(birth.boys_34)
abline(v=sum(birth1), col="red")
```



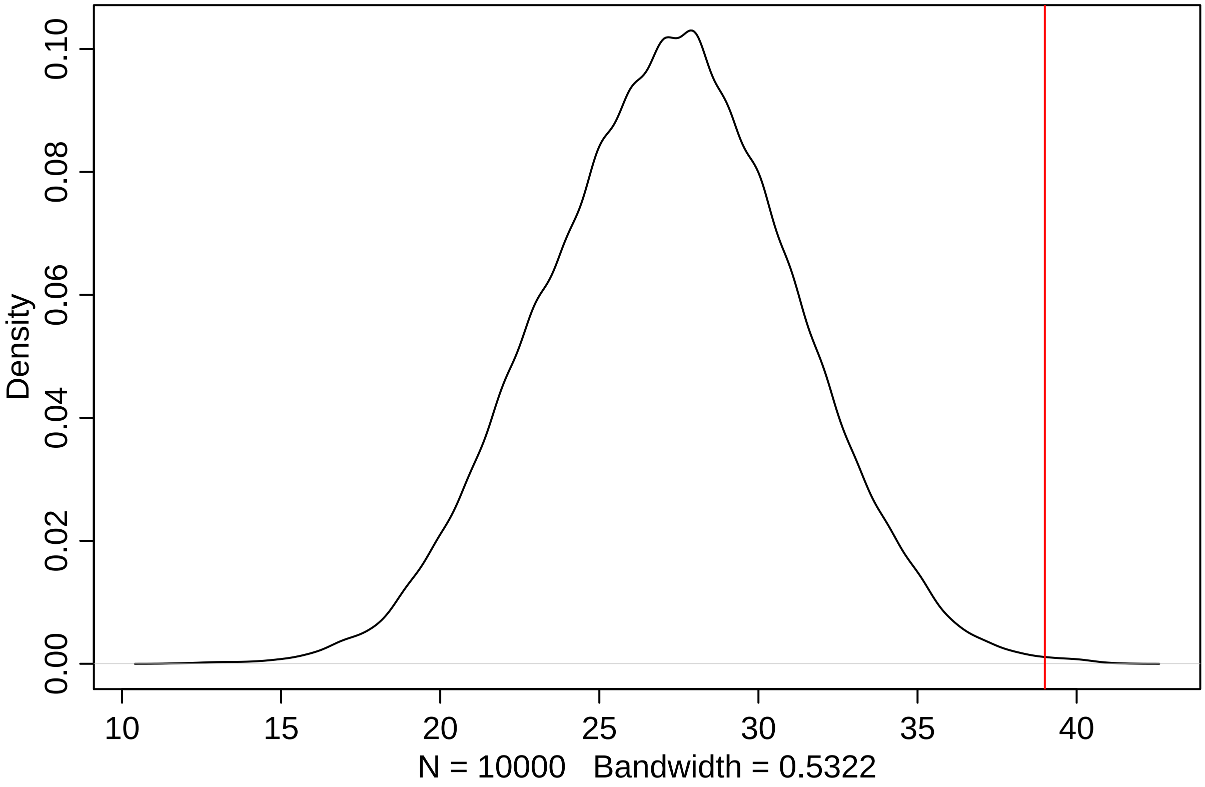
The model **overestimates** the number of first-born boys.

3H5. The model assumes the sex of the first and second births are independent. To check this assumption, focus now on second births that followed female first borns. Compare 10,000 simulated counts of boys to only second births that followed girls. To do this correctly, you need to count the number of first borns who were girls and simulate that many births, 10,000 times. Compare the counts of boys in your simulations to the actual observed count of boys following girls. How does the model look in this light? Any guesses what is going on in these data?

```
birth.first_girls <- birth1 == 0
birth.first_girls_count <- sum(birth.first_girls)
birth.boys_followed_girls <- sum(birth2[birth.first_girls])
birth.boys_35 <- rbinom(n = 1e4, size = birth.first_girls_count, prob=birth.samples)
plot(table(birth.boys_35))
abline(v=birth.boys_followed_girls, col="red")
```



```r
dens(birth.boys_35, adj=1)
abline(v=birth.boys_followed_girls, col="red")
```



The model **underestimates** the number of second born boys in the scenario (1st girl) -> (2nd boy).