

Easy

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

E1

```
sum(samples<.2)/1e4
```

```
[1] 4e-04
```

E2

```
sum(samples>.8)/1e4
```

```
[1] 0.1116
```

E3

```
sum(samples > .2 & samples < .8)/1e4
```

```
[1] 0.888
```

E4

```
quantile(samples, .2)
```

```
      20%  
0.5185185
```

E5

```
quantile(samples, .8)
```

```
      80%  
0.7557558
```

E6

```
library(rethinking)
HPDI(samples, .66)
```

```
      |0.66      0.66|
0.5085085 0.7737738
```

E7

```
quantile(samples, c(.17, .83))
```

```
      17%      83%
0.5025025 0.7697698
```

Medium

I will use stan to do this instead of of a grid approximation so that anyone who wants to can see how the stan model works. Look in the “stan_models” folder to see the model.

M1

```
library(cmdstanr)
library(rethinking)
# make the model but don't compile until checking the syntax
mod <- cmdstan_model("stan_models/c3m1.stan", compile = F)
# check the syntax
mod$check_syntax()
# syntax is good now compile the model
mod <- cmdstan_model("stan_models/c3m1.stan")

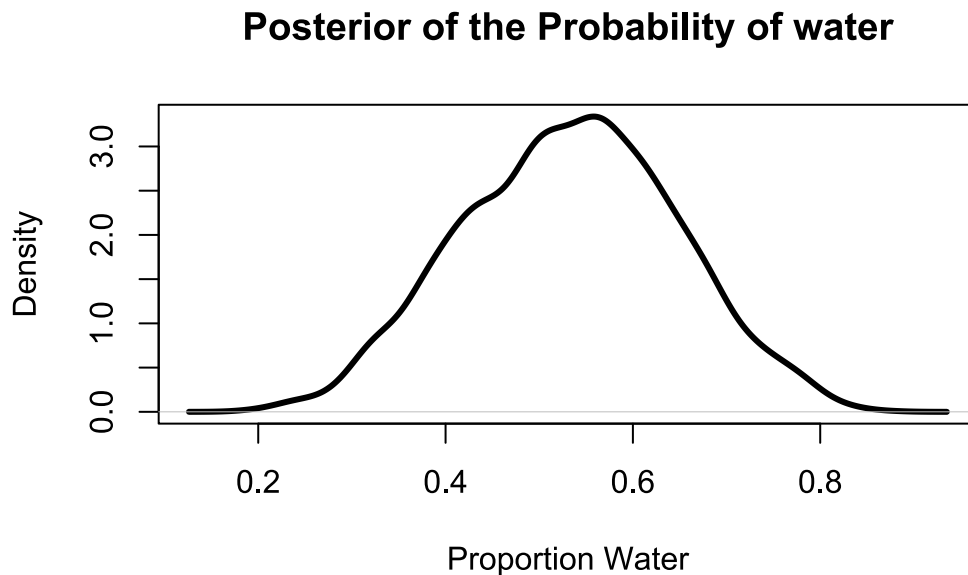
# put the data into a list
dat <- list(
  N = 1,
  y = 8,
  n = 15
)

# fit the model
fit <- mod$sample(
  data = dat,
  chains = 4,
  parallel_chains = 4,
  show_messages = F
)

# extract the posterior samples for p (probability of water)
```

```
p <- fit$draws("p", format = "df")$p

# plot the approximate posterior density
plot(density(p), lwd = 3, main = "Posterior of the Probability of water",
     xlab = "Proportion Water")
```



M2

`p` in the code block above is 4000 samples from the posterior, so I'll just resample from them. This really isn't necessary since those samples already represent the posterior, but just for fun.

```
# sample from the posterior
post_samp <- sample(p, 1e4, replace = T)

# show them both just to see it's the same.
HPDI(post_samp, .9)
```

```
|0.9    0.9|
0.351639 0.730650
```

```
HPDI(p)
```

```
|0.89   0.89|
0.35004 0.72063
```

M3

Draw a y for every value of p in the data, then we can look at the probability of getting 15.

```
sum(rbinom(4000, 15, p)==8)/4000
```

```
[1] 0.1495
```

M4

```
sum(rbinom(4000, 9, p)==6)/4000
```

```
[1] 0.18125
```

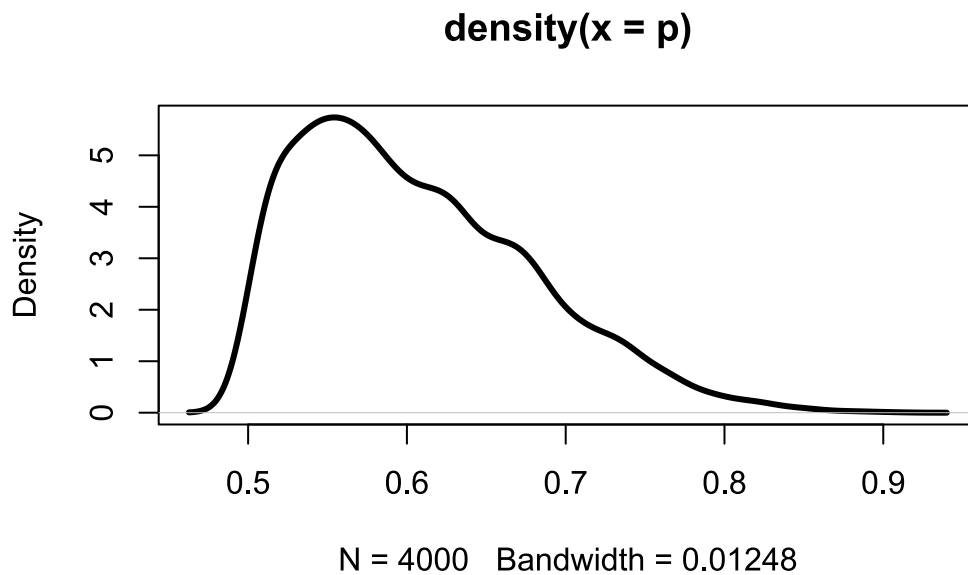
M5

```
# make the model but don't compile until checking the syntax
mod <- cmdstan_model("stan_models/c3m5.stan", compile = F)
# check the syntax
mod$check_syntax()
# syntax is good now compile the model
mod <- cmdstan_model("stan_models/c3m5.stan")

fit <- mod$sample(
  data = dat,
  chains = 4,
  parallel_chains = 4,
  show_messages = F
)

# extract the posterior samples for p (probability of water)
p <- fit$draws("p", format = "df")$p

plot(density(p), lwd = 3)
```



```
# sample from the posterior
post_samp <- sample(p, 1e4, replace = T)

# show them both just to see it's the same.
HPDI(post_samp)
```

```
|0.89    0.89|
0.501511 0.707769
```

```
HPDI(p)
```

```
|0.89    0.89|
0.500214 0.704956
```

```
sum(rbinom(4000, 15, p)==8)/4000
```

```
[1] 0.164
```

```
sum(rbinom(4000, 9, p)==6)/4000
```

```
[1] 0.2235
```

Hard

```
data(homeworkch3)
sum(birth1 + birth2)
```

```
[1] 111
```

I'll use the same stan model as for M1 above, it should work.

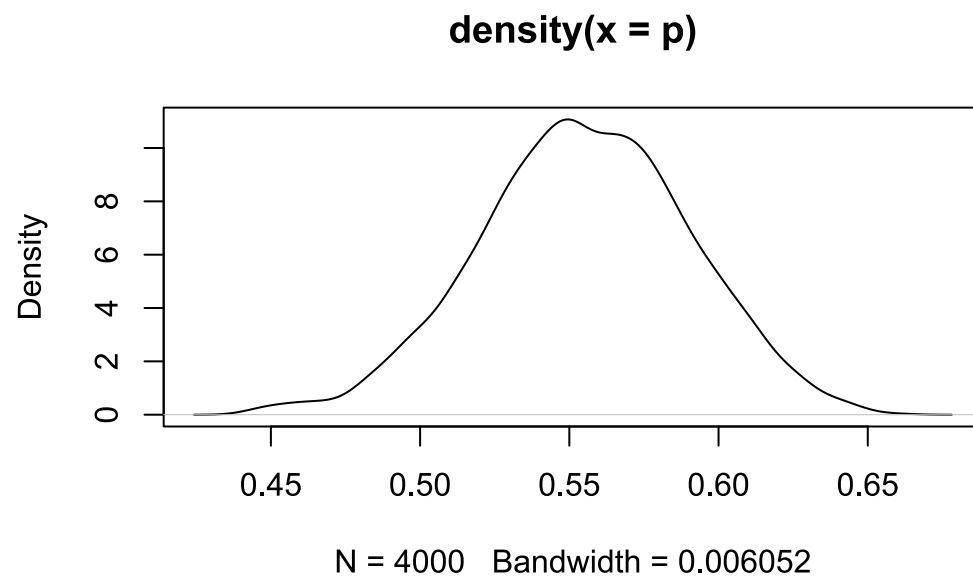
```
mod <- cmdstan_model("stan_models/c3m1.stan")

# put the data into a list
dat <- list(
  N = 1,
  y = sum(c(birth1, birth2)),
  n = length(c(birth1, birth2))
)

# fit the model
fit <- mod$sample(
  data = dat,
  chains = 4,
  parallel_chains = 4,
  show_messages = F
)

p <- fit$draws("p", format = "df")$p

plot(density(p))
```



H2

```
HPDI(p, c(50, 89, 97))
```

97	89	50	50	89	97
0.442393	0.442393	0.442393	0.659992	0.659992	0.659992