SR HW 2 3

Emilio Horner

2022-09-23

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.3.6      ✓ purrr   0.3.4
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.0      ✓ stringr 1.4.1
## ✓ readr   2.1.2      ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
```

Chapter 2

2E1. (2) Pr(rain|Monday) and (4) Pr(rain, Monday)/ Pr(Monday)

2E2.

3. The probability that it is Monday, given that it is raining.

2E3.

1. Pr(Monday|rain)
2. Pr(rain|Monday) Pr(Monday)/ Pr(rain)

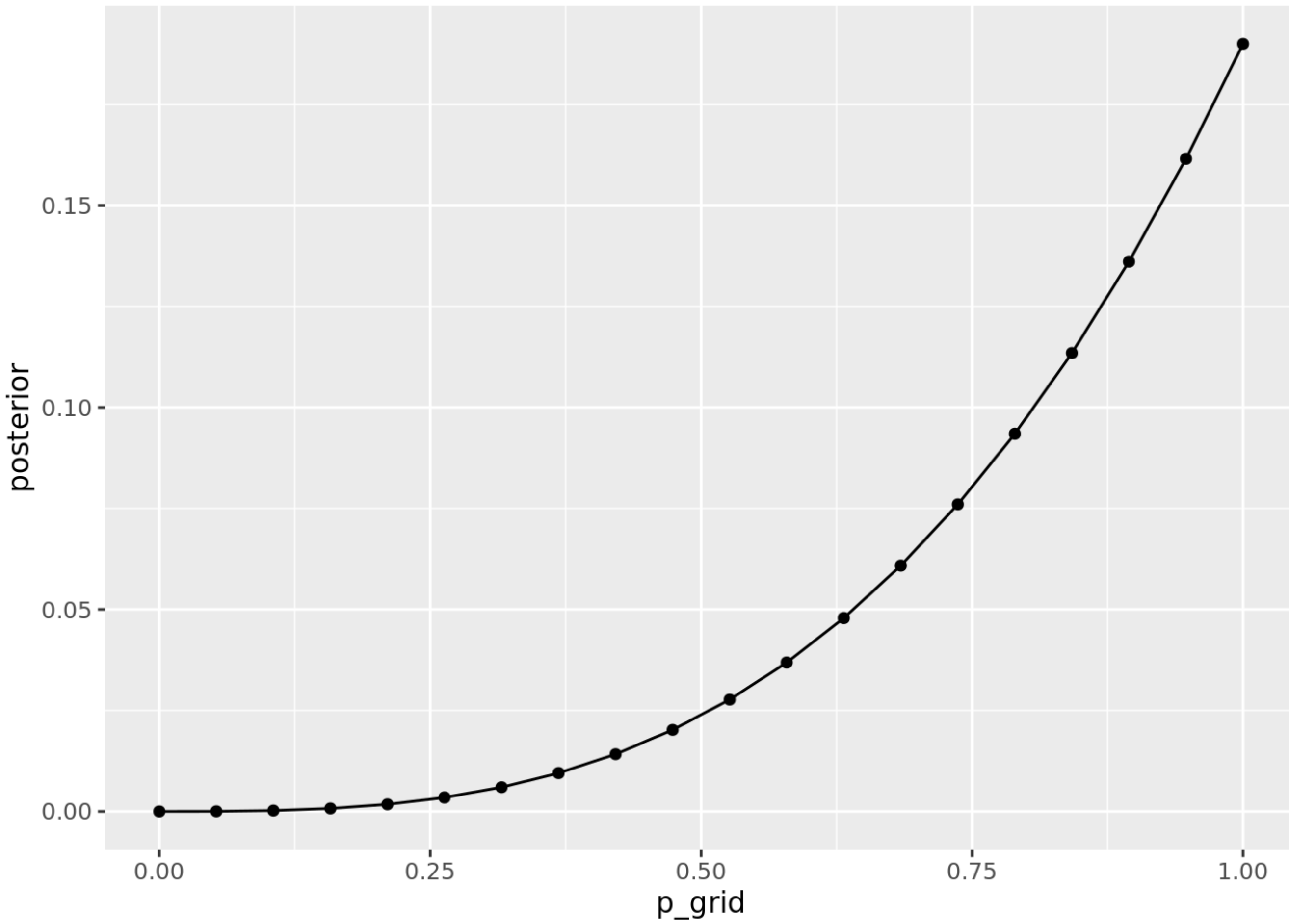
2E4.

Probability is always inaccurate in that it is not an exact map, but it is computing statistics from data that is not totalizing or 100% mapped accurately onto the real thing. For example in the globe problem, we are not actually looking at the real Earth but a model of the Earth. The world isn’t exactly 70% water. The probability .7 is just that 7/10 times we will most likely catch the globe in water according to the data so far.

2M1.

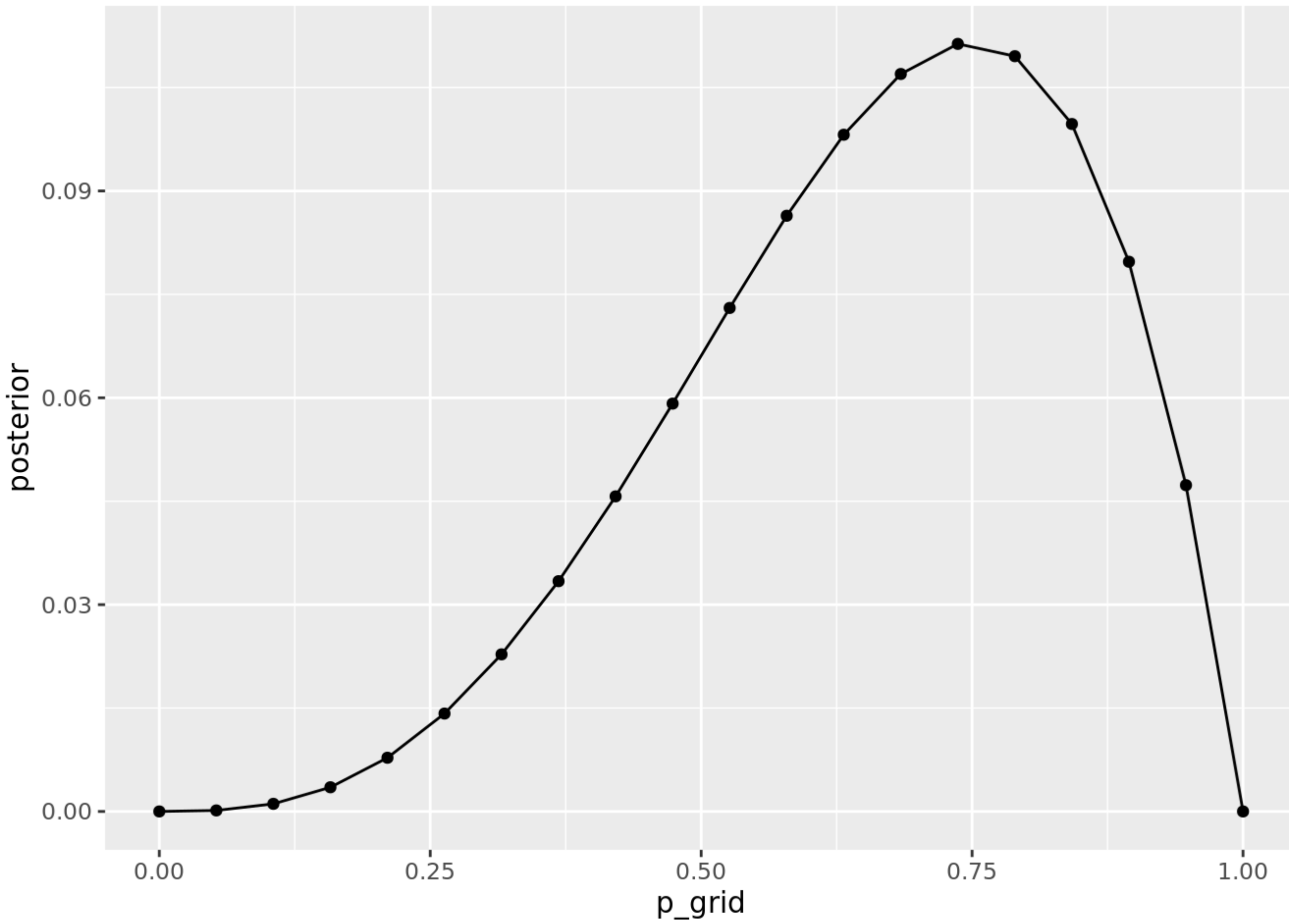
```
d <-
    tibble(p_grid = seq(from = 0, to = 1, length.out = 20),
           prior  = 1) |>
    mutate(likelihood = dbinom(3, size = 3, prob = p_grid)) |>
    mutate(unstd_posterior = likelihood * prior) |>
    mutate(posterior = unstd_posterior / sum(unstd_posterior))
```

```
 ggplot(data = d, mapping = aes(x = p_grid, y = posterior)) +
  geom_point() +
  geom_line()
```



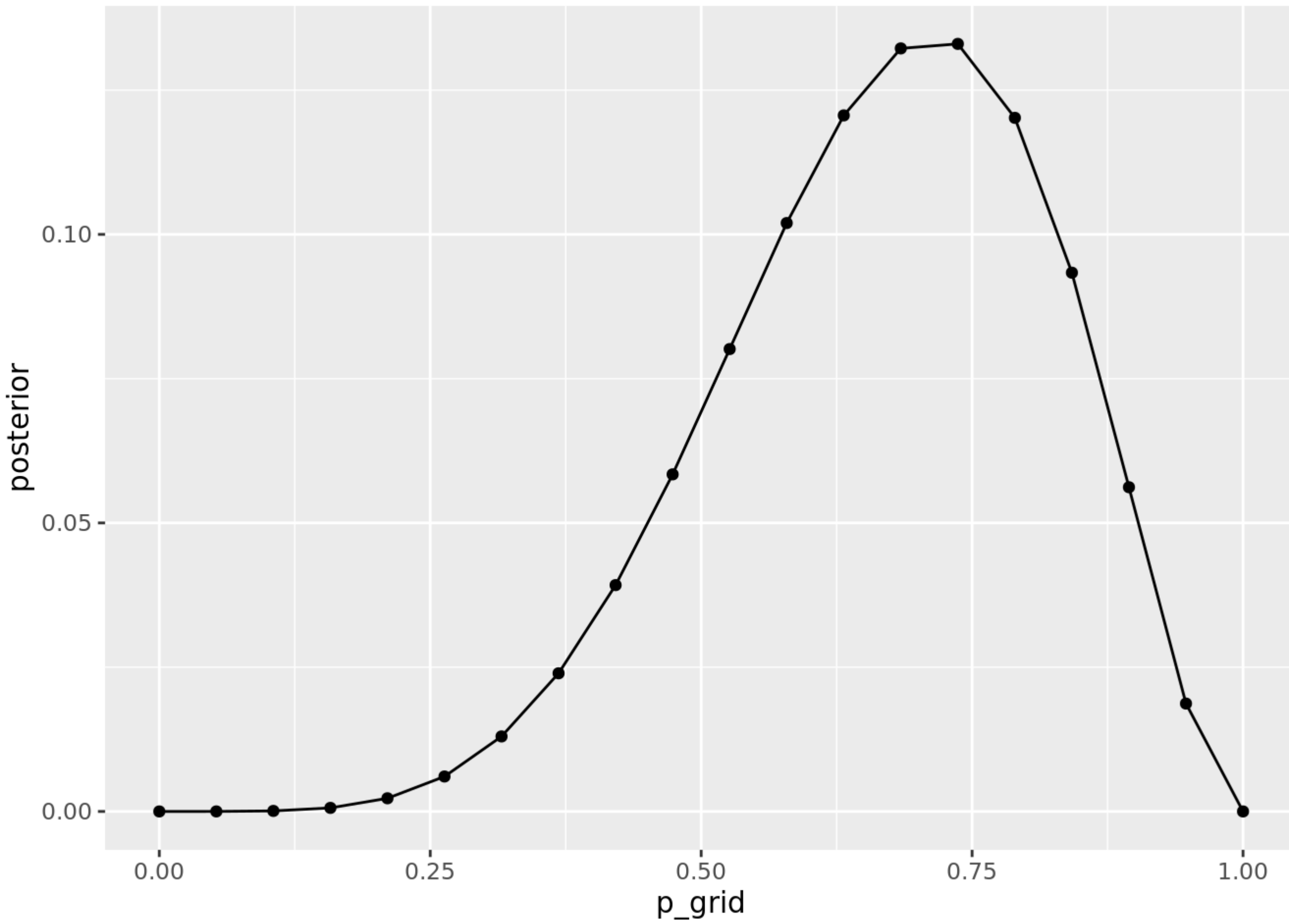

```
d <-
  tibble(p_grid = seq(from = 0, to = 1, length.out = 20),
         prior  = 1) |>
  mutate(likelihood = dbinom(3, size = 4, prob = p_grid)) |>
  mutate(unstd_posterior = likelihood * prior) |>
  mutate(posterior = unstd_posterior / sum(unstd_posterior))
```

```
ggplot(data = d, mapping = aes(x = p_grid, y = posterior)) +
 geom_point() +
 geom_line()
```




```
d <-
  tibble(p_grid = seq(from = 0, to = 1, length.out = 20),
         prior  = 1) |>
  mutate(likelihood = dbinom(5, size = 7, prob = p_grid)) |>
  mutate(unstd_posterior = likelihood * prior) |>
  mutate(posterior = unstd_posterior / sum(unstd_posterior))
```

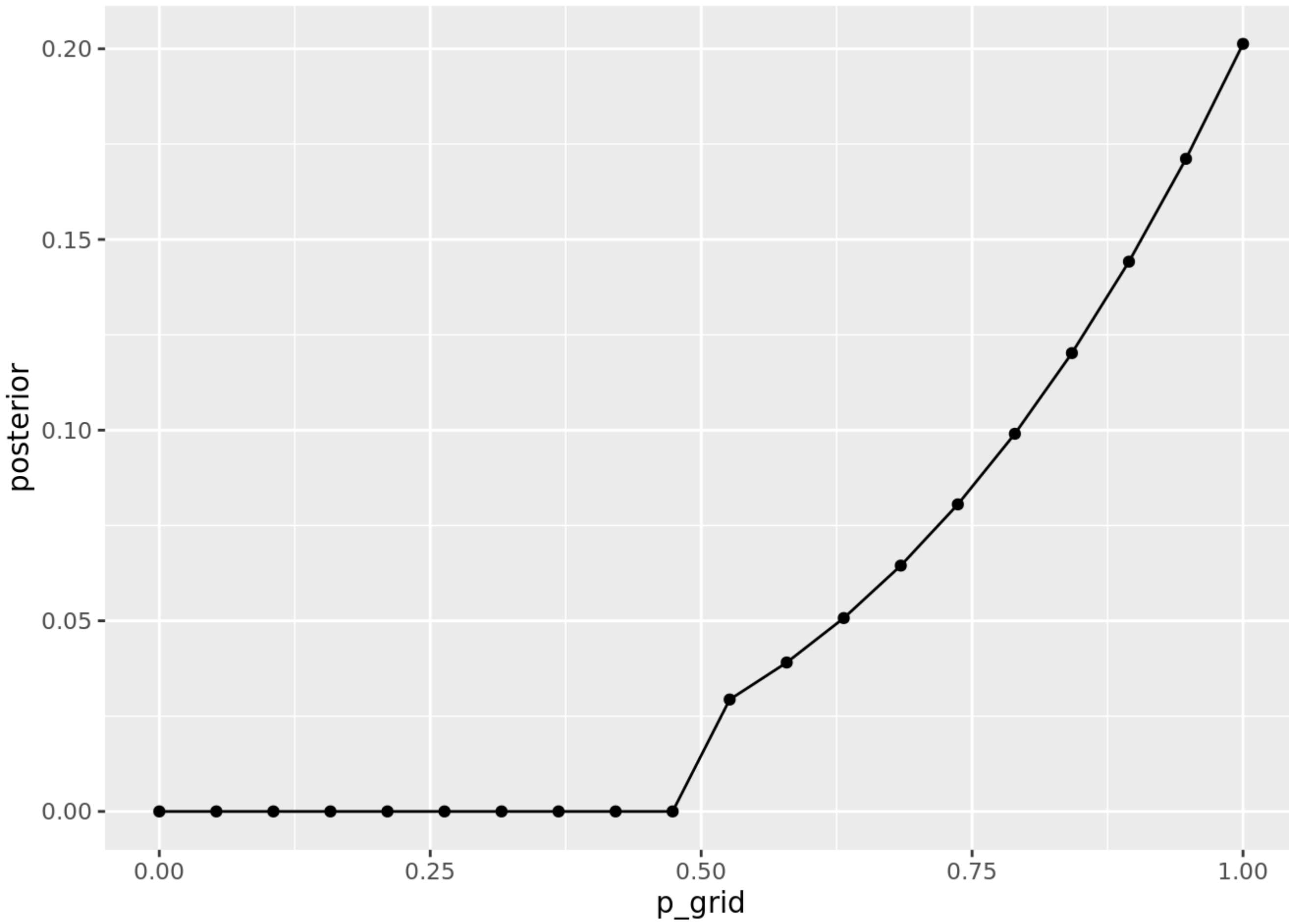
```
ggplot(data = d, mapping = aes(x = p_grid, y = posterior)) +
 geom_point() +
 geom_line()
```



2M2

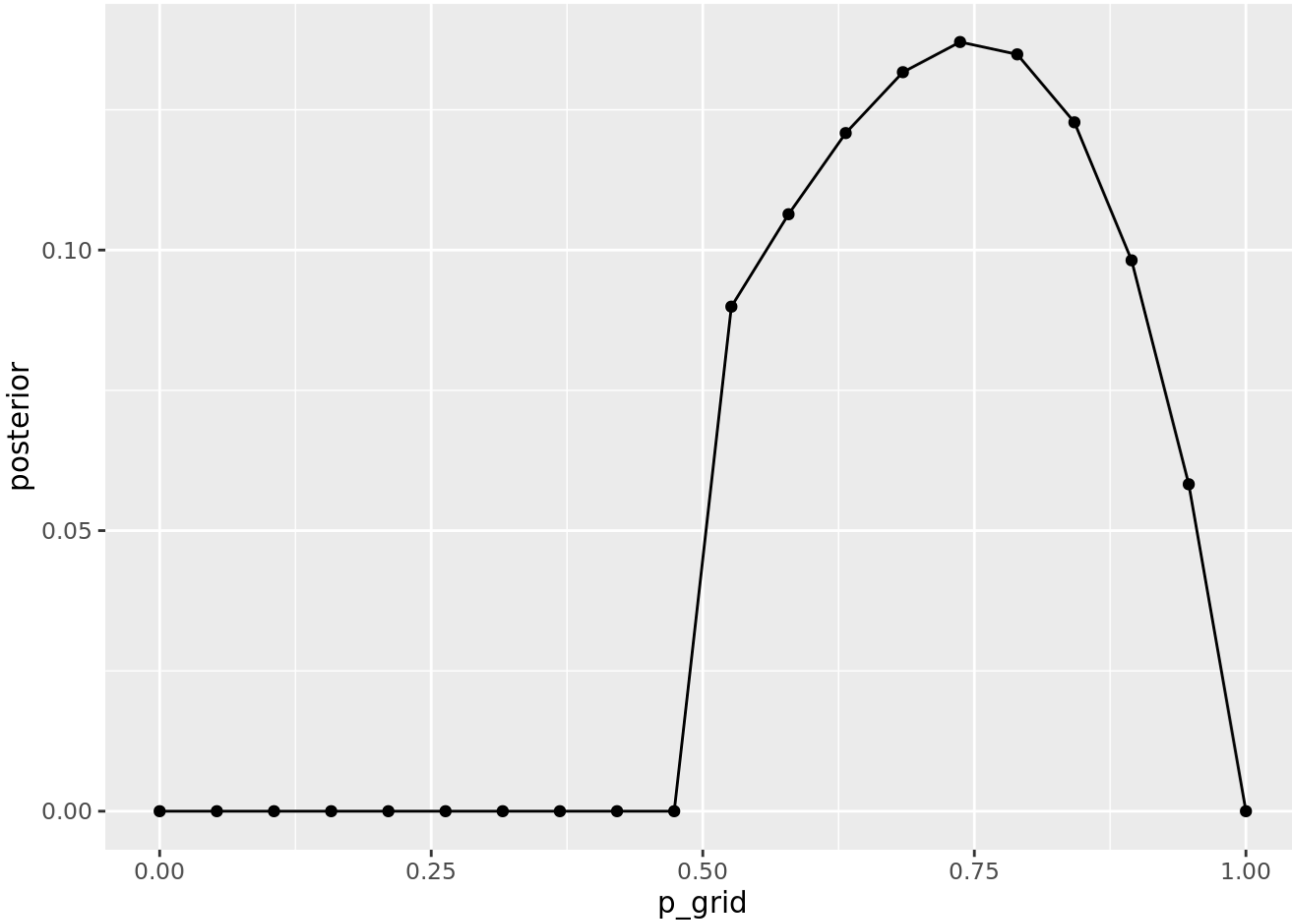
```
d <-
  tibble(p_grid = seq(from = 0, to = 1, length.out = 20),
         prior  = ifelse(p_grid < 0.5, 0, 1)) |>
  mutate(likelihood = dbinom(3, size = 3, prob = p_grid)) |>
  mutate(unstd_posterior = likelihood * prior) |>
  mutate(posterior = unstd_posterior / sum(unstd_posterior))
```

```
ggplot(data = d, mapping = aes(x = p_grid, y = posterior)) +
 geom_point() +
 geom_line()
```



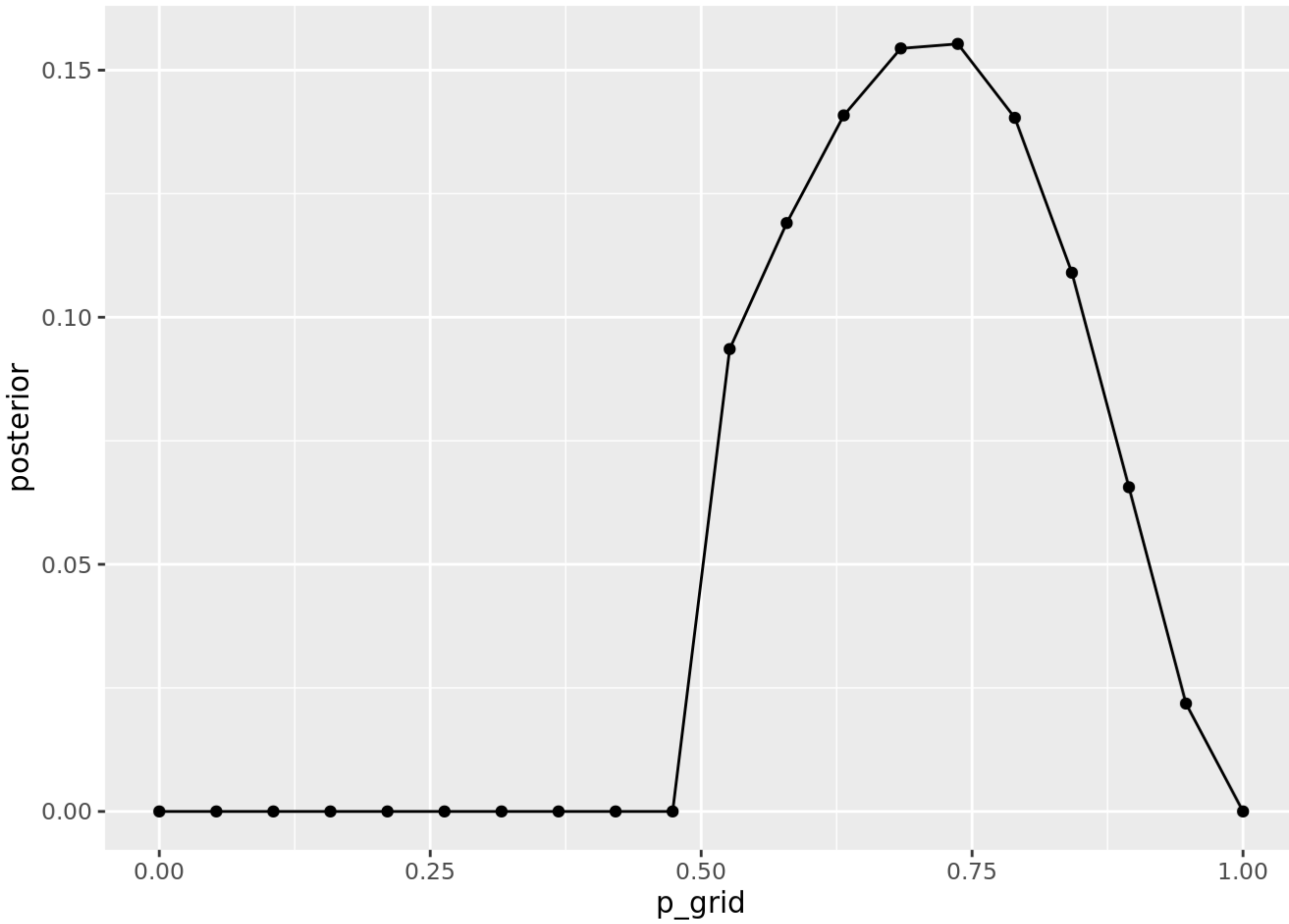

```
d <-
  tibble(p_grid = seq(from = 0, to = 1, length.out = 20),
         prior  =ifelse(p_grid < 0.5, 0, 1)) |>
  mutate(likelihood = dbinom(3, size = 4, prob = p_grid)) |>
  mutate(unstd_posterior = likelihood * prior) |>
  mutate(posterior = unstd_posterior / sum(unstd_posterior))
```

```
ggplot(data = d, mapping = aes(x = p_grid, y = posterior)) +
 geom_point() +
 geom_line()
```




```
d <-
  tibble(p_grid = seq(from = 0, to = 1, length.out = 20),
         prior  =ifelse(p_grid < 0.5, 0, 1)) |>
  mutate(likelihood = dbinom(5, size = 7, prob = p_grid)) |>
  mutate(unstd_posterior = likelihood * prior) |>
  mutate(posterior = unstd_posterior / sum(unstd_posterior))
```

```
ggplot(data = d, mapping = aes(x = p_grid, y = posterior)) +
 geom_point() +
 geom_line()
```



2M3.

Pr(land|Earth)=1−0.7=.3 Pr(land|Mars)=1 Pr(Earth)=Pr(Mars)=0.5

.3 *.5= .15

.3(.5)+1(.5)=.65

Pr(Earth|land)= .15/.65

23%

2M4.

three cards two sides each

BB, BW, WW

three ways the facing up card could be black

2/3

since there’s two b/b cards and one b/w card

2M5.

B/B, B/W, W/W, B/B

5 possible draws that could be black 4 of them would have a black card on the back, one draw would have a white draw on the back

so 4/5 times black is picked it will be B/B

2M6.

1 B/B 2 B/W 3 W/W

plausibility of p after Dnew = ways p can produce Dnew × prior plausibility p/ sum of products

2* 1/(12)+(21) =2/4 =.5

2M7.

B/B, B/W, W/W

if its black/black then there are three white options.

6/8 so .75

Chapter 3

```
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=TRUE )
```

3e1.

```
sum(samples < 0.2 ) / 1e4
```

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

0.0004

3e2.

```
mean(samples > 0.8)
```

```
## [1] 0.1116
```

3e3.

```
mean( samples > 0.2 & samples < 0.8 )
```

```
## [1] 0.888
```

3e4.

```
quantile( samples, probs = 0.2 )
```

```
##       20%
## 0.5185185
```

3e5.

```
quantile( samples, probs = 0.8 )
```

```
##       80%
## 0.7557558
```

```
library(rethinking)
```

```
## Loading required package: rstan
```

```
## Loading required package: StanHeaders
```

```
## rstan (Version 2.21.5, GitRev: 2e1f913d3ca3)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
##
## Attaching package: 'rstan'
```

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

```
## Loading required package: cmdstanr
```

```
## This is cmdstanr version 0.5.3
```

```
## - CmdStanR documentation and vignettes: mc-stan.org/cmdstanr
```

```
## - Use set_cmdstan_path() to set the path to CmdStan
```

```
## - Use install_cmdstan() to install CmdStan
```

```
## Loading required package: parallel
```

```
## rethinking (Version 2.23)
```

```
##
## Attaching package: 'rethinking'
```

```
## The following object is masked from 'package:rstan':
##
##     stan
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```
## The following object is masked from 'package:stats':
##
##     rstudent
```

3e6.

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

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

3e7.

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

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

3m1.

8 water, 15 total tosses

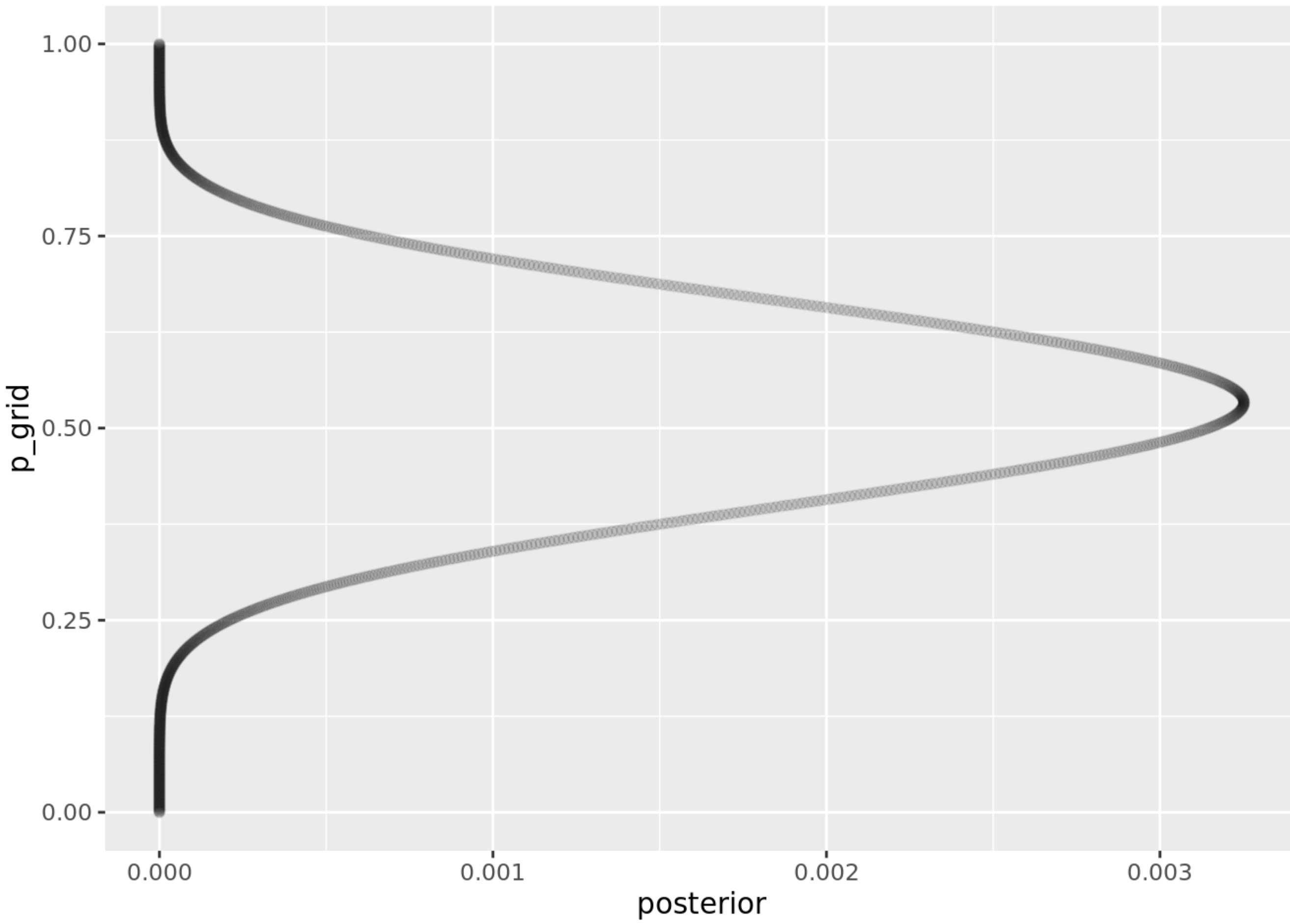
posterier distribution using grid approximation

```
# how many grid points would you like?
n <- 1000
n_success <- 8
n_trials  <- 15

(
  d <-
  tibble(p_grid = seq(from = 0, to = 1, length.out = n),
         # note we're still using a flat uniform prior
         prior  = 1) %>%
  mutate(likelihood = dbinom(n_success, size = n_trials, prob = p_grid)) %>%
  mutate(posterior = (likelihood * prior) / sum(likelihood * prior))
)
```

```
## # A tibble: 1,000 × 4
##     p_grid prior likelihood posterior
##      <dbl> <dbl>      <dbl>     <dbl>
##  1 0           1   0         0
##  2 0.00100     1   6.44e-21  1.03e-22
##  3 0.00200     1   1.64e-18  2.62e-20
##  4 0.00300     1   4.17e-17  6.67e-19
##  5 0.00400     1   4.13e-16  6.62e-18
##  6 0.00501     1   2.45e-15  3.92e-17
##  7 0.00601     1   1.04e-14  1.67e-16
##  8 0.00701     1   3.56e-14  5.70e-16
##  9 0.00801     1   1.03e-13  1.65e-15
## 10 0.00901     1   2.62e-13  4.20e-15
## # … with 990 more rows
```

```
ggplot(data = d, mapping =aes(x = posterior, y = p_grid)) +
  geom_point(alpha = 1/10)
```



3m2.

```
samples <- sample( p_grid, prob=posterior, size=1e4, replace=TRUE)
HPDI(samples, prob = 0.9 )
```

```
##      |0.9      0.9|
## 0.4184184 0.8678679
```

3m3.

rbinom(n_draws, size = n, prob = probability)

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

```
mean( x == 8 )
```

```
## [1] 0.1085
```

3m4.

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

```
## [1] 0.1959
```

3m5.

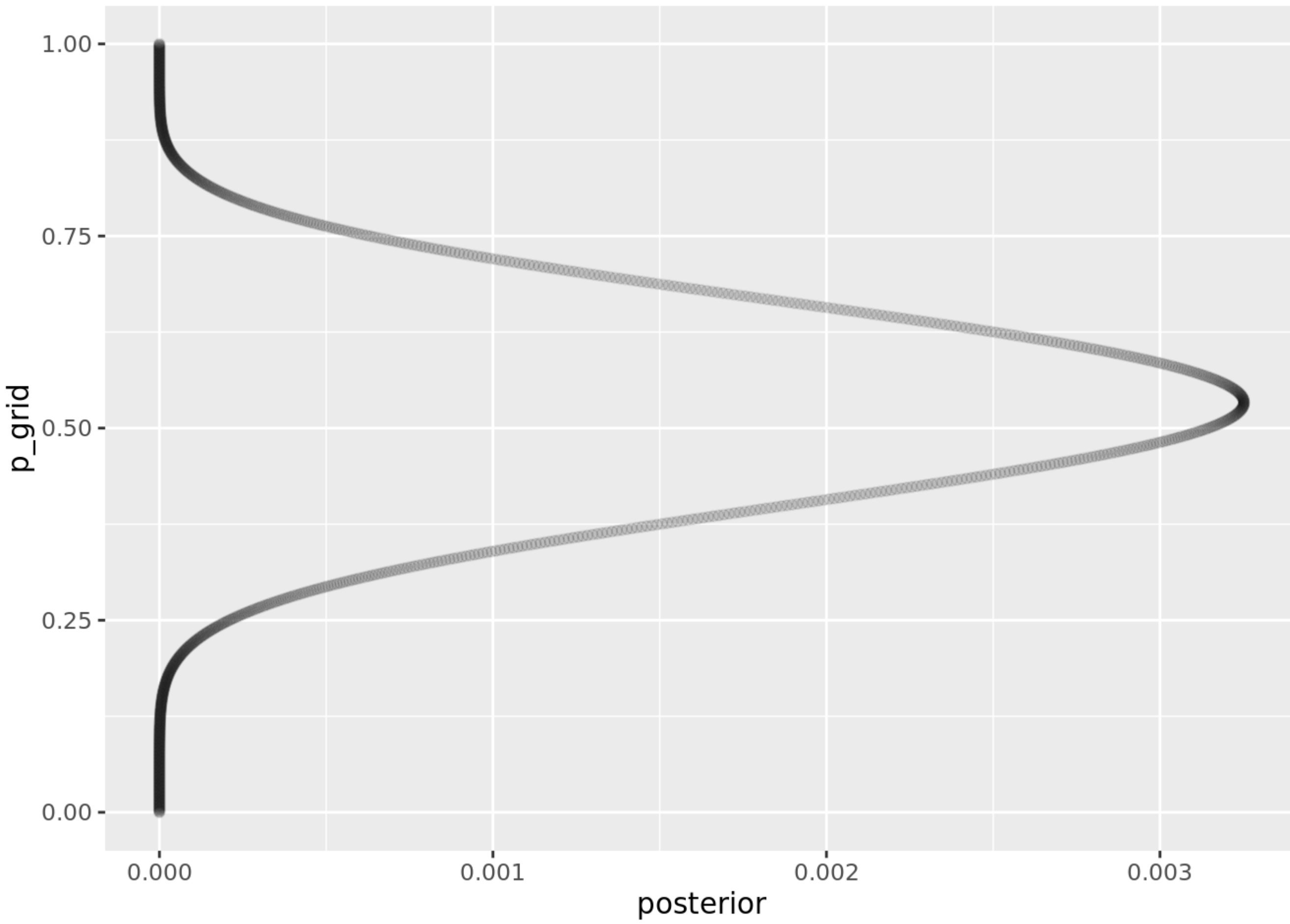
use a prior that is zero below p = 0.5 and a constant above p = 0.5

```
# how many grid points would you like?
n <- 1000
n_success <- 8
n_trials  <- 15

(
  d <-
  tibble(p_grid = seq(from = 0, to = 1, length.out = n),
         informed_prior <- ifelse( p_grid < 0.5, 0, 2 ) ) |>
  mutate(likelihood = dbinom(n_success, size = n_trials, prob = p_grid)) |>
  mutate(posterior = (likelihood * prior) / sum(likelihood * prior))
)
```

```
## # A tibble: 1,000 × 4
##     p_grid `informed_prior <- ifelse(p_grid < 0.5, 0, 2)` likelihood posterior
##      <dbl>                                          <dbl>      <dbl>     <dbl>
##  1 0                                                    0   0         0
##  2 0.00100                                              0   6.44e-21  1.03e-22
##  3 0.00200                                              0   1.64e-18  2.62e-20
##  4 0.00300                                              0   4.17e-17  6.67e-19
##  5 0.00400                                              0   4.13e-16  6.62e-18
##  6 0.00501                                              0   2.45e-15  3.92e-17
##  7 0.00601                                              0   1.04e-14  1.67e-16
##  8 0.00701                                              0   3.56e-14  5.70e-16
##  9 0.00801                                              0   1.03e-13  1.65e-15
## 10 0.00901                                              0   2.62e-13  4.20e-15
## # … with 990 more rows
```

```
ggplot(data = d, mapping =aes(x = posterior, y = p_grid)) +
  geom_point(alpha = 1/10)
```




```
samples <- sample( p_grid, prob=posterior, size=1e4, replace=TRUE)
HPDI(samples, prob = 0.9 )
```

```
##      |0.9      0.9|
## 0.4104104 0.8598599
```

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

```
mean( x == 8 )
```

```
## [1] 0.1098
```

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

```
## [1] 0.2077
```

the prior serves to update the information we have going into calculating the posterior and the distribution