Stat 341 – Homework 02

Luca Seazzu

April 11, 2023

```
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)
```

SR 3E1

```
mean(samples < 0.2)
```

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

SR 3E2

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

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

SR 3E3

```
1 - mean(samples < 0.2) - mean(samples > 0.8)
```

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

SR 3E4

```
quantile(samples, 0.20)
```

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

SR 3E5

```
quantile(samples, 1 - 0.20)
```

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

SR 3E6

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

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

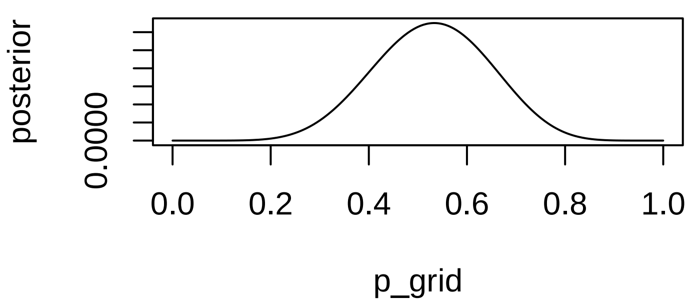
SR 3E7

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

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

SR 3M1

```
p_grid <- seq(from = 0, to = 1, length.out = 1e3)
prior <- rep(1, 1e3)
likelihood <- dbinom(8, size = 15, prob = p_grid)
posterior <- likelihood * prior
posterior <- posterior / sum(posterior)
plot(x = p_grid, y = posterior, type = "l")
```



SR 3M2

```
samples <- sample(p_grid, prob = posterior, size = 1e4, replace = TRUE)
HPDI(samples, prob = .90)
```

```
##      |0.9      0.9|
## 0.3293293 0.7167167
```

N1

```
movies <- read.csv('https://sldr.netlify.app/data/movielens.csv')

# display a table of your variable of interest
```

```r
library(mosaic)
# inputs are the variable you want to tally up, like: ~ VARIABLE_NAME
# and the name of the dataset, data  = ____
tally(~ animation, data  = movies) |>
  # adding kable() formats the table more prettily
  knitr::kable()
```

animation	Freq
Animated	158
Not Animated	1787

```r
# create a dataset containing the individual observations, if you only have n trials and n successes
n_failures <- 1787
n_successes <- 158
own_data <- tibble(observations =
  c(rep('animated', n_successes),
  rep('not animated', n_failures)))
# all the successes will be first in the list, then the failures after
head(own_data)
```

```
## # A tibble: 6 x 1
##   observations
##   <chr>
## 1 animated
## 2 animated
## 3 animated
## 4 animated
## 5 animated
## 6 animated
```

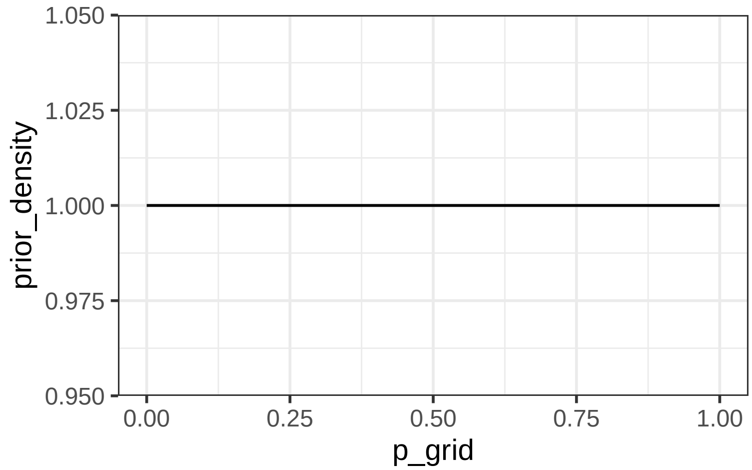
Part A

The quantity to estimate is if the movie is animated or not.

Part B

I have little no no knowledge of the movie database so I am going to use an uninformative prior with a uniform distribution.

```r
my_unif_prior <- tibble(p_grid = seq(from = 0, to = 1, length.out = 1000),
                        prior_density =dunif(p_grid, min = 0, max = 1))
gf_line(prior_density ~ p_grid, data = my_unif_prior)
```



Part C

```
grid_movie_model <-
  tibble(p_grid = seq(from = 0, to = 1, length.out = 100000),      # define grid
         prior = dunif(p_grid, min = 0, max = 1)
         ) |>                                             # define prior
  mutate(likelihood = dbinom(n_successes,
                             size = n_successes  + n_failures,
                             prob = p_grid)) |>  # compute likelihood at each value in grid
  mutate(unstd_posterior = likelihood * prior) |>             # compute product of likelihood and prior
  mutate(posterior = unstd_posterior / sum(unstd_posterior))   # standardize the posterior, so it sums


# to peek at the results table
glimpse(grid_movie_model)
```

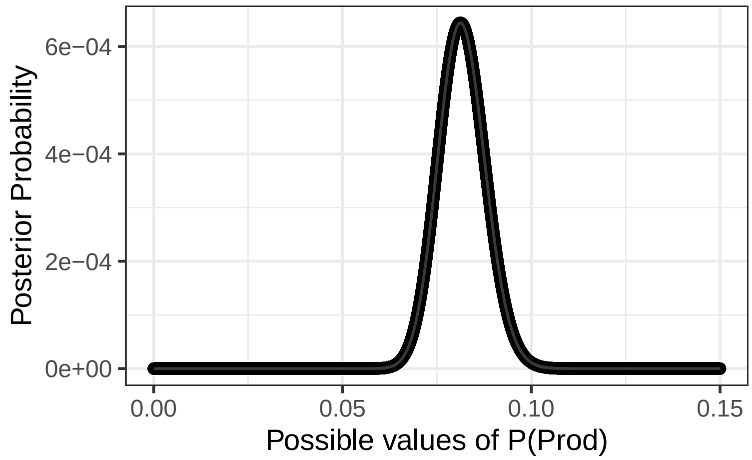
```
## Rows: 100,000
## Columns: 5
## $ p_grid          <dbl> 0.0000000000, 0.0000100001, 0.0000200002, 0.0000300003~
## $ prior           <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ likelihood      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ unstd_posterior <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ posterior       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
```

```
grid_post_plot <- gf_point(posterior ~ p_grid,
                           data = grid_movie_model) |>
  # this is optional -- adds a line in addition to  the dots
  gf_line(color = 'grey44', alpha = 0.5) |>
  gf_labs(x = 'Possible values of P(Prod)',
          y = 'Posterior Probability')

grid_post_plot |> gf_lims(x = c(0, 0.15))
```

```
## Warning: Removed 85000 rows containing missing values (`geom_point()`).

## Warning: Removed 85000 rows containing missing values (`geom_line()`).
```



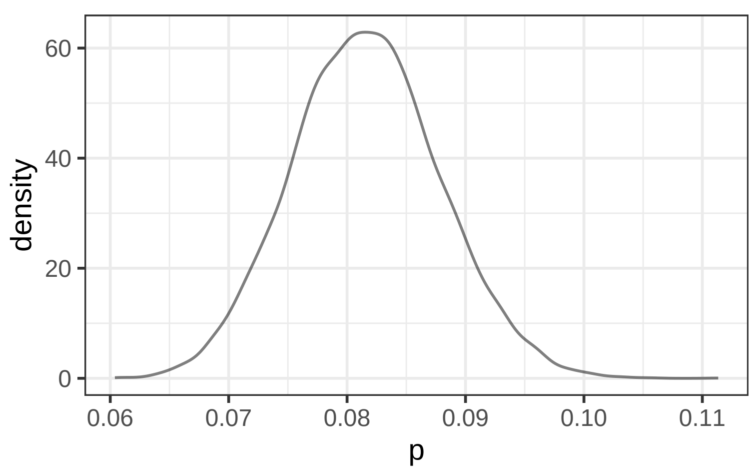

```
# zooms the graph in
```

Part D

```
post_samp_size = 10000
# eliminate sampling variation: get the same exact samples each time you run/knit
set.seed(7)
grid_post_samp <- grid_movie_model |>
  slice_sample(n = post_samp_size,
               weight_by = 10000*posterior,
               replace = TRUE) |>
  # sort of optional - but to avoid confusion, keep ONLY the posterior
  select(p_grid) |>
  rename(p = p_grid)
```

```
gf_dens(~p, data = grid_post_samp)
```

```
## Warning: `stat(density)` was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
```



The first way I chose to represent the posterior was a density line graph. I chose this one because it is the best way to represent a sample from the posterior and see its distribution and where different likelihoods are.

```r
HPDI(grid_post_samp$p, prob  = 0.95)
```

```
##      |0.95      0.95|
## 0.06989070 0.09421094
```

The second way I chose to represent the posterior was numerical and showing the Highest Posterior Density Interval which gives the boundaries of the .95 proportion of the sample from the posterior that has the highest density. In this case it lies between roughly .069 and .094 which looks similar to the results of the graph but in numerical form.

Part E

I learned from this that based on the movie database data, there is a range of probabilities that the movie is animated and within a 95% bounderie of the proportion, lies between 6.98% and 9.42% that a movie is animated vs. not animated.

Part F

Some qualms would first be the data source since 1945 movies seems kind of low as a data size to make interpretations. It would be nice to also see if there are more kids movies that are animated versus not kids movies since that is probably heavily involved. I do like the way a bayesian model presents the findings though.