Assignment 2

Natali Tckvitishvili

2024-06-04

General Information

- **Points**: Assignment 2 comprises of 6 tasks, 2 points each (12 in total). 2 points are obtained for complete and correct answers. 1 point is obtained for a proper approach or if only part of the task is solved.
- **Submission**: Hand in the assignment as a `Markdown` report (RMarkdown or Quarto) rendered as PDF. The PDF report should show the result(s), the code that produced the result(s), and possibly additional text or comment. Also indicate your name. The report should be uploaded on Moodle until Wednesday, June 5, 9:45 am.
- **Working in teams**: Everyone needs to hand in a report on Moodle. However, the report can be handed in as a team work (max. 2 people). When working in teams, state at the beginning of the document, who you worked with. It Ideally, teams use GitHub and add a link to the GitHub repository to which both contributed.
- **Code**: To automate code wrapping (such that long code lines are not cut off), install the formatR package and add the following code chunk at the beginning of the document:

```
knitr::opts_chunk$set(tidy = TRUE, tidy.opts=list(width.cutoff=50))
```

```r
library(stats)
library(dplyr)
library(ggplot2)
```

Task Set 1

For tasks 1.1-1.2, suppose there are 3 companies, Company A to C. Company A has a customer satisfaction rate of .70, Company B of .50, and Company C of .80. Further suppose that you receive 10 customer reviews (6 positive, 4 negative) for the same company, but you don't know for which company. Assume that Company B is twice as likely to obtain reviews than Company A and C.

```r
# satisfaction rates
sf_A <- 0.7
sf_B <- 0.5
sf_C <- 0.8

pos_0 <- 6 # number of initial positive reviews
neg_0 <- 4 # number of initial negative reviews
n_0 <- pos_0 + neg_0 # total initial reviews

freq_B <- 2 # B is twice as likely to receive reviews
```

Task 1.1

Show that the posterior probability that Company A was rated is ≈ 0.29.

To calculate prior probability for company A, we need to use the knowledge that company B is twice likely to obtain reviews than others.

$$prior_A + prior_B (= 2 * prior_A) + prior_C = 1 => prior_A = 1/4$$

```r
# priors for each company
prior_A <- 1/4
prior_B <- 1/2
prior_C <- 1/4

# data frame of priors and satisfaction rates (aka candidates)
companies_0 <- data.frame(
  company = c("A", "B", "C"),
  prior = c(prior_A, prior_B, prior_C),
  satisfaction = c(sf_A, sf_B, sf_C)
)

# posterior function
posterior <- function(data, pos, n) {
  likelihood <- dbinom(pos, size = n, prob = data$satisfaction)
  posterior <- likelihood * data$prior
  posterior_norm <- posterior / sum(posterior) # normalization
  data.frame(data[1], posterior_norm) # 1 column is a company
}
```

```r
posterior_0 <- posterior(companies_0, pos_0, n_0)
filter(posterior_0, company == "A")$posterior_norm
```

```
## [1] 0.2865594
```

As we can see, the posterior for company A is 0.2865594 which is approximately 0.29

Task 1.2

Suppose you receive 10 more reviews (9 positive and 1 negative). Show that the posterior probability that Company C received the reviews increases by ≈ 33 percentage points, when considering all 20 rather than only the first 10 reviews. To obtain the updated posterior, compute the likelihood of the 10 most recent reviews only.

```r
pos_1 <- 9 # new positive reviews
neg_1 <- 1 # new negative reviews
n_1 <- pos_1 + neg_1

companies_1 <- data.frame(
  company = c("A", "B", "C"),
  prior = posterior_0$posterior_norm,
  satisfaction = c(sf_A, sf_B, sf_C)
)

posterior_1 <- posterior(companies_1, pos_1, n_0)
filter(posterior_1, company == "C")$posterior_norm - filter(posterior_0, company == "C")$posterior_norm
```

```
## [1] 0.329651
```

The change is 0.3296651 which is approximately 33 percentage points.

Task Set 2

For tasks 2.1 and 2.2, suppose there are Factory A and Factory B, producing the same product. The company C receives equally many shipments from both factories. Even though the machines, processes, and standards are virtually identical, the factories differ in their defect rates. Shipments from Factory A entail defective products 10% of the time, shipments from Factory B entail defective products 20% of the time.

```
defect_A <- 0.1
defect_B <- 0.2

# probability of getting shipment from each factory is the same, so:
prior_A <- 0.5
prior_B <- 0.5
```

Task 2.1

You receive a shipment from one of the factories, and upon inspection, you find that the shipment contains defective products. Compute the probability that the next shipment from this company will also contain defective products.

First we need to compute probability of getting defective products from both factories - these will be our next priors We use Bayes' theorem:

$$P(A|defect) = P(defect|A) * P(A)/P(defect)$$
$$P(B|defect) = P(defect|B) * P(B)/P(defect)$$

```
# total probability of getting defective product
p_defect <- defect_A * prior_A + defect_B * prior_B

# probability that defective product is from A
p_A_defect <- defect_A * prior_A / p_defect

# probability that defective product is from B
p_B_defect <- defect_B * prior_B / p_defect
```

Now we'll use p_A_defect = 0.3(3) and p_B_defect = 0.6(6) as priors to calculate probability of obtaining defective products in the next shipment

```
# probabilities are already normalized so we don't need to divide them by total

# probability ob obtaining defective products again from company A
p_A_defect_next <- defect_A * p_A_defect

# probability ob obtaining defective products again from company B
p_B_defect_next <- defect_B * p_B_defect

# probabilities are independent, so we can sum them up to obtain total
p_defect_next <- p_A_defect_next + p_B_defect_next
p_defect_next
```

```
## [1] 0.1666667
```

Therefore, the probability of obtaining defective products again from the same company is ~16.7%

Task 2.2

Suppose the R&D department came up with a Machine Learning algorithm that (imperfectly) identifies the factory based on the shipped products. But the classification algorithm is imperfect. This is the information you have about the algorithm:

- The probability it correctly identifies a Factory A product is 93%.
- The probability it correctly identifies a Factory B product is 87%.

When applying the the algorithm to the shipped products, the test is positive for Factory A. Including the defect data from 2.1, compute the posterior probability that your shipment is from Company A.

Our prior is p_A_defect = 0.3(3) as it's the probability that defective product comes from company A Likelihood is 93%, as it shows the probability that algorithm is right given that it is actually from A

To compute marginal likelihood, we need to take probability that algorithm detected A This happens in 93% cases when it's actually from A and in 13% when it wrongly detected it as B Therefore, marginal likelihood is 0.93 * p_A_defect + 0.13 * p_B_defect

```
ident_A <- 0.93
ident_B <- 0.87

# marginal likelihood
p_algorithm <- ident_A * p_A_defect + (1 - ident_B) * p_B_defect

# probability that shipment is from A
p_A_defect * ident_A / p_algorithm
```

```
## [1] 0.7815126
```

So, the probability that shipment is from company A is approximately 78%

Task Set 3

For Task 3.1 and 3.2, suppose, one last time, you want to estimate the proportions of land on the earth's.

Task 3.1

Specify a prior distribution and store 10,000 random samples from it in a vector `sample`. Plot the prior distribution and briefly explain your choice of the prior.

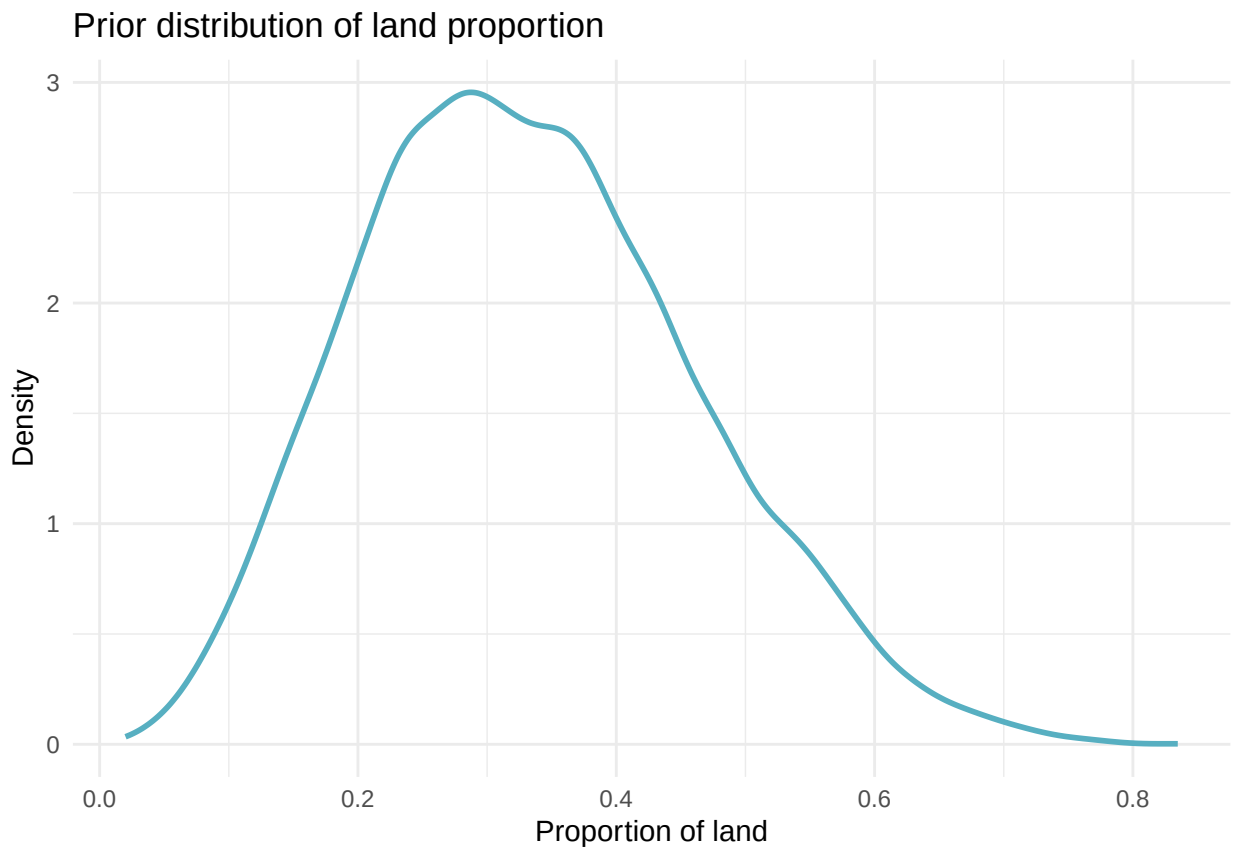
We want to estimate a proportion, therefore it is a good idea to choose beta-distribution To select a and b parameters I played a bit, looks like these are pretty adequate :)

```
a <- 4
b <- 8

theta <- seq(0, 1, length.out = 1000) # 1000 values
d <- dbeta(theta, shape1 = a, shape2 = b)
summary <- data.frame(theta, d)

# sample from prior
set.seed(70)
n <- 10000
prior_sample <- data.frame(sample = rbeta(n, a, b))

ggplot(summary) +
  geom_density(data = prior_sample, aes(x = sample), color = "#57afc1", size = 1) +
  theme_minimal() +
  labs(
    title = "Prior distribution of land proportion",
    x = "Proportion of land",
    y = "Density")
```



Task 3.2

Run the following code chunk that uses your object sample to obtain prior probabilities for the possible proportions of land 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1 that approximate your prior distribution.

```
prop <- seq(0, 1, length.out = 12)
priors <- vector("numeric", length(prop))
for (i in seq_along(prop)){
  priors[i] <- round(sum(prior_sample >= prop[i] & prior_sample < prop[i+1]) / 1e4, 2)
}
poss <- data.frame(prop_L = seq(0, 1, .1), prior = priors[1:11])
```

Use these priors to compute the posterior probability after observing 26 times land in 100 globe tosses. Take 1,000 samples from the posterior distribution and with each sample, predict the outcome of 100 globe tosses. Plot the posterior predictions in a histogram.

```
num_tosses <- 100
land <- 26

# posterior function
posterior_globe <- function(data_post, land, n) {
  likelihood <- dbinom(land, size = n, prob = data_post$prop_L)
  posterior <- likelihood * data_post$prior
  posterior_norm <- posterior / sum(posterior) # normalization
```

```r
  data.frame(data_post, lh = round(likelihood, 3), post = round(posterior_norm, 3))
}

posterior <- posterior_globe(poss, land, num_tosses)

set.seed(70)
posterior_sample <- data.frame(sample = sample(posterior$prop_L, size = 1000, replace = TRUE, prob = pos

num_outcomes <- 100
predictions <- data.frame(count = rbinom(n = num_outcomes, size = num_outcomes, prob = posterior_sample$

predictions %>% ggplot(aes(x = count)) +
  geom_histogram(fill = "#d6d444", color = "black", alpha = .5, bins = 100) +
  scale_x_continuous(limits = c(0, num_outcomes), breaks = seq(0, num_outcomes, 10)) +
  labs(
    title = "Posterior predictions of land count",
    x = "Number of simulated L out of 100",
    y = "Frequency"
  )
```