BDA - Assignment 1

Anonymous

15 9 2019

Contents

Used libraries:

```
library(dplyr)
library(ggplot2)
library(aaltobda)
library(knitr)
```

Exercise 1

Explain the following terms with one sentence:

- probability: A numerical mapping from some random *outcome* $->$ $p \in [0, 1]$, representing the proportion of the outcome in some *set* of oucomes.
- probability mass: Probability of a discrete (random) event/outcome.
- probability density: The speed at which probability mass is acquired as the range of continuous outcome(s) widens.
- probability mass function (pmf): Function that maps values of discrete random variables to corresponding probabilities.
- probability density function (pdf): Function that maps values ($a \leq x \leq b$) of continuous random variables to probability density - derivative of cdf.
- probability distribution: Describes how probabilities are distributed for some random, discrete or continuous, outcome (event/variable/etc.).
- discrete probability distribution: Same as pmf
- continuous probability distribution: Same as pdf
- cumulative distribution function (cdf): Integral of pdf, which describes how probability mass grows as probability density is summed over an interval (can be similarly defined as a sum for discrete random variables).
- likelihood: quantifies how likely (probable) data is, *given* some set of parameters.

Exercise 2

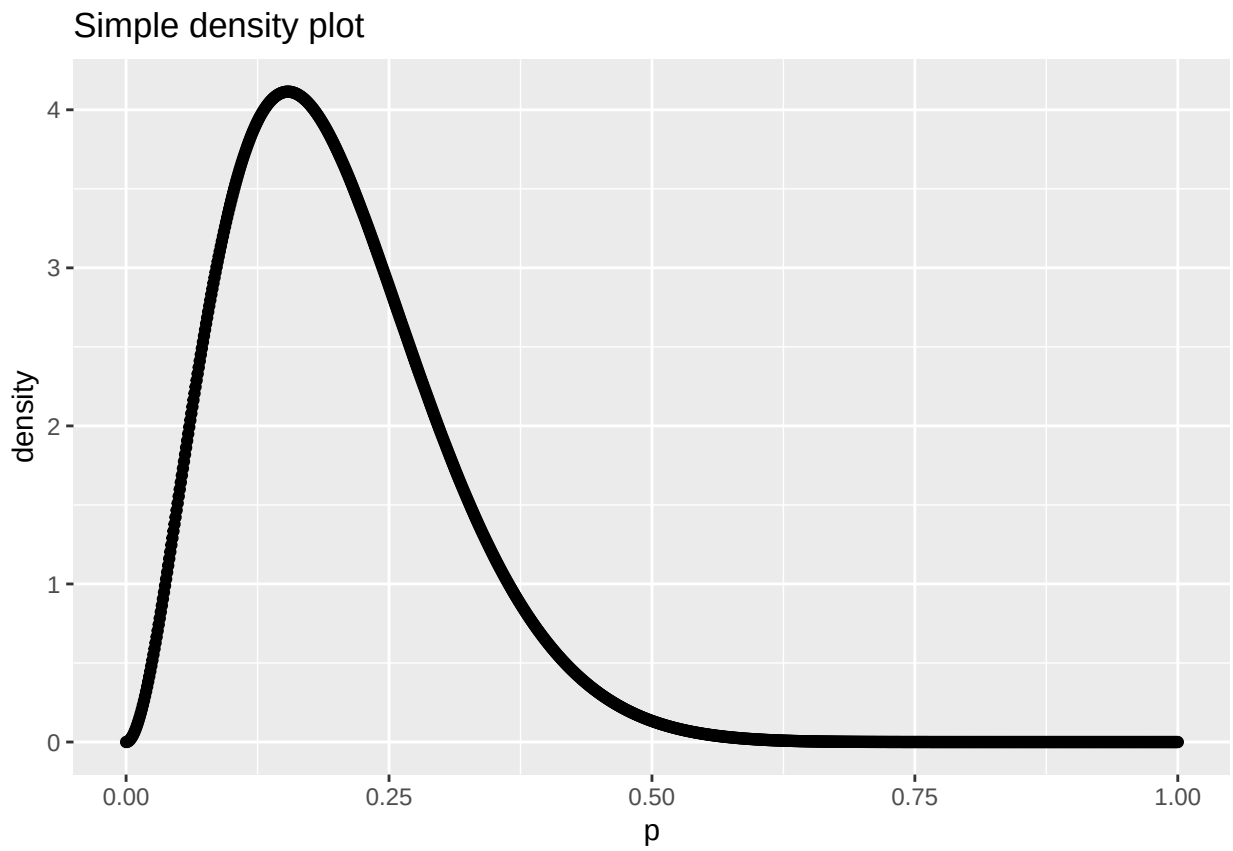
a)

Plot the density function of Beta-distribution, with mean $\mu = 0.2$ and variance $\sigma^2 = 0.01$. The parameters α and β of the Beta-distribution are related to the mean and variance according to the following equations:

$$\alpha = \mu(\frac{\mu(1-\mu)}{\sigma^2} - 1)$$

$$\beta = \frac{\alpha(1-\mu)}{\mu}$$

```
#helper functions
Alpha <- function(mu, sigsq) {

  mu*(mu*(1-mu)/sigsq-1)

}

Beta <- function(mu, alpha) {

  alpha*(1-mu)/mu

}


mu <- .2
sigsq <- .01

A <- Alpha(mu, sigsq)
B <- Beta(mu, A)


q <- seq(from=0, to=1, length.out = 1000)
dp <- dbeta(q, A, B)


qplot(seq(0,1,length.out = 1000), dp, main = 'Simple density plot', ylab = 'density', xlab='p')
```



b,c,d)

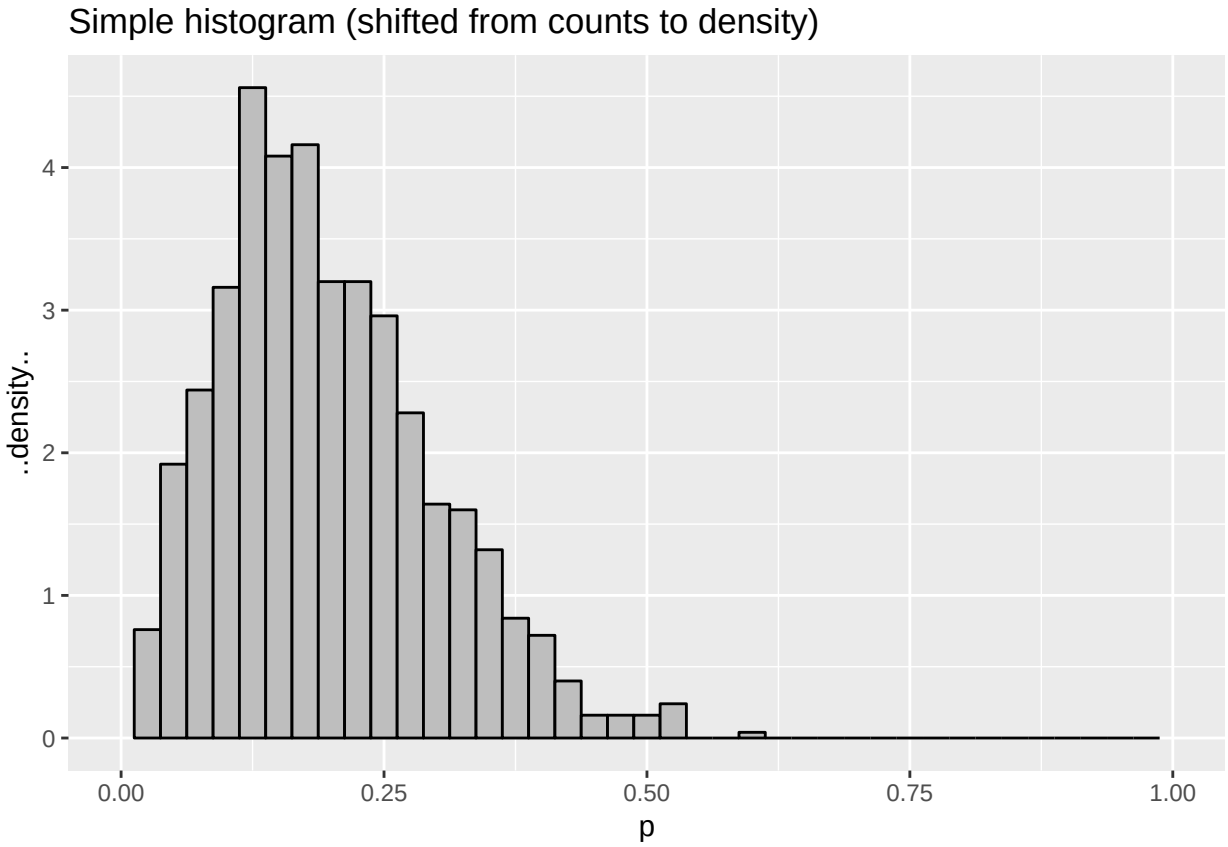
(b)Take a sample of 1000 random numbers from the above distribution and plot a histogram of the results. Compare visually to the density function.

(c)Compute the sample mean and variance from the drawn sample. Verify that they match (roughly) to the true mean and variance of the distribution.

(d)Estimate the central 95%-interval of the distribution from the drawn samples.

I am solving these all here in one go, and present a more robust graph to display the results in the end

```
# for b)
r <- rbeta(1000, A, B)
qplot(r, geom = 'histogram',y = ..density.., binwidth=0.025,
      fill=I('gray'), color=I('black'),main = 'Simple histogram (shifted from counts to density)',
      xlim = c(0,1), xlab = 'p')
```



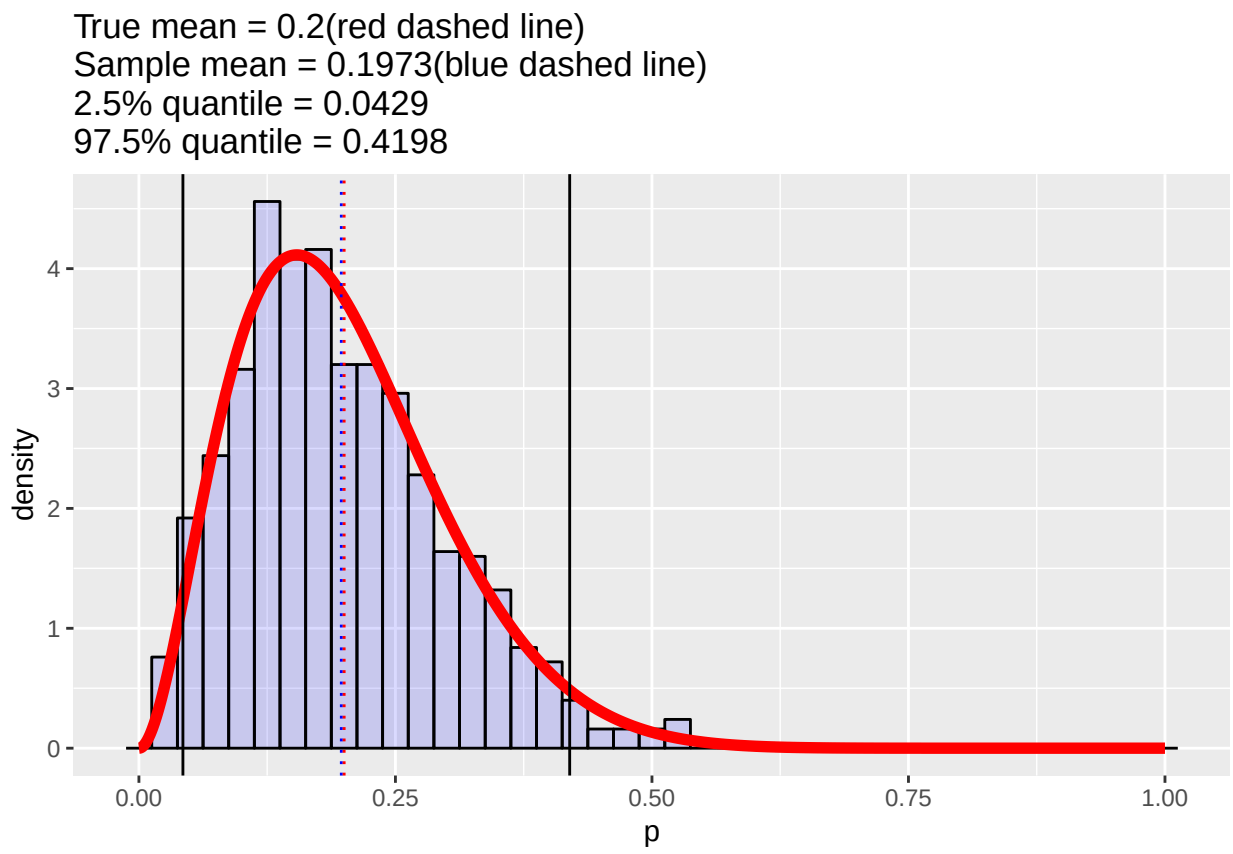

```
# for c)
mu_sample <- mean(r)
var_sample <- var(r)

table <- matrix(c(mu, mu_sample, sigsq, var_sample), nrow = 2, byrow = T) %>% as.data.frame()
colnames(table) <- c('True','Sample')
rownames(table) <- c('mean','var')

table %>% kable()
```

	True	Sample
mean	0.20	0.1972578
var	0.01	0.0102527

```
# for d)
plot_q <- quantile(r, probs = c(0.025, 0.975))
plot_q %>% kable(., col.names = 'p')
```

	p
2.5%	0.0428740
97.5%	0.4198373

```r
# Plot for all b,c,d)
ds <- data.frame(p = r, x = q)

ggplot(data = ds, aes(p)) +
  geom_histogram(aes(y = ..density..), binwidth = 0.025, color='black', fill='blue', alpha=0.15) +
  geom_line(aes(x=q, y=dp), color = 'red', size=2) +
  geom_vline(xintercept = mean(r), linetype = 'dotted', color = 'blue', size= 0.5) +
  geom_vline(xintercept = mu, linetype = 'dotted', color = 'red', size = 0.5) +
  geom_vline(xintercept = plot_q[1], size = .5) +
  geom_vline(xintercept = plot_q[2], size = .5) +
  labs(title=paste('True mean = ',mu, '(red dashed line)',
                   '\nSample mean = ', round(mean(r), digits=4), '(blue dashed line)',
                   '\n2.5% quantile = ', round(plot_q[1], digits=4),
                   '\n97.5% quantile = ', round(plot_q[2],digits = 4),sep=''))
```



Exercise 3

A group of researchers has designed a new inexpensive and painless test for detecting lung cancer. The test is intended to be an initial screening test for the population in general. A positive result (presence of lung cancer) from the test would be followed up immediately with medication, surgery or more extensive and expensive test. The researchers know from their studies the following facts:

- Test gives a positive result in 98% of the time when the test subject has lung cancer.
- Test gives a negative result in 96 % of the time when the test subject does not have lung cancer.
- In general population approximately one person in 1000 has lung cancer.

The researchers are happy with these preliminary results (about 97% success rate), and wish to get the test to market as soon as possible. How would you advise them? Base your answer on elementary probability calculus.

My approach: I will condition the given test probabilities (neg., pos.) with the population priors (healthy, sick) := (no cancer, cancer). First, I'll define some data structures:

```
health <- c('healthy', 'sick')   # (no cancer, cancer)
test <- c('neg.', 'pos.')        # test labels

P_healthy <- 0.999
P_sick <- 0.001

# Un-conditioned probabilities:

data <- matrix(c(0.96, 0.04, 0.02, 0.98), nrow = 2, byrow = T,
               dimnames = list(health, test)) %>% as.data.frame()

kable(data)
```

	neg.	pos.
healthy	0.96	0.04
sick	0.02	0.98

Now, of utmost importance would be to know what proportion of positive test results would actually come from healthy people - and with the same trouble, how many come from truly sick people:

$$Pr(healthy|pos.) = \frac{Pr(healthy)Pr(pos.|healthy)}{Pr(pos.)} = \frac{Pr(healthy)Pr(pos.|healthy)}{\sum_{healthy}^{sick} Pr(*)Pr(pos.|*)}$$

```
P_positive <- P_healthy*data['healthy', 'pos.'] + P_sick*data['sick', 'pos.']

P_healthy_pos <- P_healthy*data['healthy', 'pos.'] / P_positive
P_sick_pos <- 1 - P_healthy_pos
```

And just to get a good overall picture of the robustness of the test, we can calculate the other conditional probabilities in a similar manner:

$$Pr(sick|neg.) = \frac{Pr(sick)Pr(neg.|sick)}{Pr(neg.)} = \frac{Pr(sick)Pr(neg.|sick)}{\sum_{healthy}^{sick} Pr(*)Pr(neg.|*)}$$

```
P_negative <- P_sick*data['sick', 'neg.'] + P_healthy*data['healthy','neg.']

P_sick_neg <- P_sick*data['sick','neg.'] / P_negative
P_healthy_neg <- 1 - P_sick_neg
```

And finally let's table the conditional probabilities:

```
df <- matrix(c(P_healthy_neg, P_healthy_pos, P_sick_neg, P_sick_pos),
             nrow = 2, byrow = T,
             dimnames = list(health, test)) %>% as.data.frame()

kable(df)
```

	neg.	pos.
healthy	0.9999791	0.9760625
sick	0.0000209	0.0239375

Now we can easily read from the table, what is the share of the positive test results:

Out of all positive test results, 0.9760625 ($\approx 97.6\%$) are false positives. I.e., only 2.4% are actually true positives. Based on this, it would be highly inappropriate to start medication, not to mention surgery, solely based on this test. In (this) state of ignorance, only acceptable way forward would be to give patients a more extensive test in the case they have tested initially positive.

Caviat: Given the specifics of the test, and the nature of the (assumed) randomness in the test results, there might be different ways to solve these formentioned challenges. For example, should the uncertainty in the test be of non-determinist nature, i.e., truly random (aleatoric), it would be possible to make the test more effective just by repeating it. On the other hand, if the uncertainty is of epistemic nature it might still be possible to find the underlying factors responsible for the false positives (e.g., some medications or hereditory traits might effect false positive rate). And if the effects of those factors could be properly quantified (modelled), they could be incorporated into the test; thus making it more robust. So, further study might still increase the true accuracy of the test, making it a sufficient screener even for medication/surgery.

Exercise 4

We have three boxes, A, B, and C. There are:

- 2 red balls and 5 white balls in the box A,
- 4 red balls and 1 white ball in the box B, and
- 1 red ball and 3 white balls in the box C.

Consider a random experiment in which one of the boxes is randomly selected and from that box, one ball is randomly picked up. After observing the color of the ball it is replaced in the box it came from. Suppose also that on average box A is selected 40% of the time and box B 10% of the time.

a) What is the probability of picking a red ball?

Probability of *red from* $\mathsf{Box_A}$ is the probability of picking red from box A times the probability of picking box A - and similarly for Boxes B,C. Hence, probability of picking a red ball can be calculated by summing conditional probabilities

$$Pr(Red|\mathsf{Box_i})Pr(\mathsf{Box_i}), i \in \{A, B, C\}$$

Below is an implementation of the function that calculates the result:

```
#Boxes
A <- c(2,5)
B <- c(4,1)
C <- c(1,3)

boxes <- rbind(A,B,C) %>% as.matrix()
rownames(boxes) <- c('A','B','C')
colnames(boxes) <- c('Red','White')


p_red <- function(data) {
  #Picks a box, and then red | box
  #Constant probabilities: P(A)=0.4, P(B)=0.1, P(C)=0.5
  priors <- list(A=0.4, B=0.1, C=0.5)
```

```
  priors$A*data['A','Red']/sum(data['A',]) +
    priors$B*data['B','Red']/sum(data['B',]) +
    priors$C*data['C','Red']/sum(data['C',])

}
boxes %>% kable()
```

	Red	White
A	2	5
B	4	1
C	1	3

```
P_red <- p_red(boxes)

print(paste('The probability of picking a red ball:', P_red))
```

```
## [1] "The probability of picking a red ball: 0.319285714285714"
```

So the result is 0.3192857

b) If a red ball was picked, from which box it most probably came from?

So practically we are calculating now

$$Pr(\text{Box}_i|\text{Red}) = \frac{Pr(Red|\text{Box}_i)Pr(\text{Box}_i)}{Pr(Red)}$$

And we can use p_red(...) implemented abowe. Here is a function to do it:

```
p_box <- function(data) {
  # P(Box|Red) = P(Red|Box)P(Box) / P(Red)
  priors <- c(0.4, 0.1, 0.5)
  weights <- apply(data[,c('Red','White')],1, sum)

  ret <- priors * data[,'Red'] / weights
  ret/p_red(data)
}
#using same boxes
P_boxes <- p_box(boxes)
```

So the results are:

	Pr(Box | Red)
A	0.3579418
B	0.2505593
C	0.3914989

, and the most likely box (given *red*) is C

Exercise 5

Assume that on average fraternal twins (two fertilized eggs) occur once in 150 births and identical twins (single egg divides into two separate embryos) once in 400 births (Note! This is not the true values, see Exercise 1.5 in BDA3). American male singer-actor Elvis Presley (1935 { 1977) had a twin brother who died in birth. What is the probability that Elvis was an identical twin? Assume that an equal number of boys and girls are born on average.

This is pretty straight-forward calculation as long as we pay attention to all possible combinations of pairs of siblings (given twin "type").

- Possible pairs of identical twins: {(B,B), (G,G)} <=> same egg, same gender.
- Possible pairs of fraternal twins: {(B,B),(B,G),(G,B),(G,G)} <=> two eggs, possibly two genders (*see an alternative approach at the end*)

$$Pr(id\_twins|\{B,B\}) = \frac{Pr(\{B,B\}|id\_twins)Pr(id\_twins)}{Pr(\{B,B\})}$$

Where

$$Pr(\{B,B\}) = \sum_{id->frat} Pr(\{B,B\}|twin\_type)$$

Here is a function implementation for the probability of Elvis having had an identical brother:

```
p_identical_twin <- function(fraternal_prob, identical_prob) {
  #P(id_twin | brothers) = P({B,B} | id_twins)*P(id_twins) / P({B,B})
  # where P({B,B}) = P({B,B} | id_twins)*P(id_twins) + P({B,B} | frat_twins)*P(frat_twins)

  Prob_frat_brothers <- 0.25 # = Prob_frat_sisters   (symmetric)
  Prob_id_brothers <- 0.50  # ---"----

  P_brothers <- Prob_id_brothers*identical_prob + Prob_frat_brothers*fraternal_prob

  #return:
  Prob_id_brothers*identical_prob / P_brothers

}

#ASKED:

Elvis_result <- p_identical_twin(1/150, 1/400)
Elvis_result
```

```
## [1] 0.4285714
```

The probability that Elvis' brother was an identical twin is 0.4285714

NOTE: There is at least one other way to calculate this, by considering only pairs {B,B}, {B,G} for fraternal twins and lone pair {B,B} for identical twins. This approach uses the knowledge of Elvis being a boy directly, and result is obviously the same. However, I thought the first way was in some sense more general (i.e., the reasoning could be easily expanded to cover more cases).