

# BDA - Assignment 2

Anonymous

## Contents

### Inference for binomial proportion

1

```
# To install aaltobda, see the General information in the assignment.
```

```
remotes::install_github("avehtari/BDA_course_Aalto", subdir = "rpackage", upgrade = "never")
```

```
## Skipping install of 'aaltobda' from a github remote, the SHA1 (38f34d35) has not changed since last
```

```
## Use `force = TRUE` to force installation
```

### Inference for binomial proportion

```
# Installing libraries and setting up dataset
```

```
library(aaltobda)
```

```
data("algae")
```

```
# to check if the code is running correctly (i.e., has been programmed correctly)
```

```
algaeTest <- c(0,1,1,0,0,0)
```

a) and b)

The following is my work:

```
beta_point_est <- function(prior_alpha,
                           prior_beta,
                           data) {
  n <- length(data)
  y <- sum(data)
  E_pay <- (prior_alpha + y)/(prior_alpha + prior_beta + n)
  return(E_pay)
}
```

```
beta_interval <- function(prior_alpha,
                           prior_beta,
                           data,
                           prob) {
  n <- length(data)
  y <- sum(data)

  E_pay <- beta_point_est(prior_alpha = 2,
                           prior_beta = 10,
                           data = data)
```

```
Var_pay <- (E_pay*(1-E_pay)) / (prior_alpha + prior_beta + n + 1)
```

a) Formulate (1) the likelihood  $P(Y|\pi)$  as a function of  $\pi$ , (2) the prior  $p(\pi)$ , and (3) the resulting posterior  $P(\pi|Y)$ . Report the posterior in the format  $\text{Beta}(\cdot, \cdot)$  where you replace  $\cdot$ 's with the correct numerical values.

Denominator:

$$P(Y) = \int_0^1 \binom{n}{y} \pi^y (1-\pi)^{n-y} d\pi = \frac{1}{n+1}$$

Prior:

$$P(\theta) = \text{Beta}(\pi | \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha) \cdot \Gamma(\beta)} \cdot \pi^{\alpha-1} \cdot (1-\pi)^{\beta-1}$$

Likelihood:

$$P(Y|\theta) = \text{Bin}(Y|n, \pi) = \binom{n}{y} \pi^y (1-\pi)^{n-y}$$

Bayes:

$$P(\theta|Y) = (n+1) \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha) \cdot \Gamma(\beta)} \cdot \binom{n}{y} \cdot \pi^{\alpha+y-1} (1-\pi)^{\beta+n-y-1}$$

$$\Rightarrow P(\theta|Y) \propto \text{Beta}(\theta | \alpha+y, \beta+n-y)$$

$$\Rightarrow E(\theta|Y) = \frac{\alpha+y}{\alpha+\beta+n} \quad \text{beta-point-est}(\cdot)$$

$$\Rightarrow \text{Var}(\theta|Y) = \frac{(\alpha+y)(\beta+n-y)}{(\alpha+\beta+n)^2(\alpha+\beta+n+1)} \quad \text{For beta-interval}(\cdot)$$

Figure 1: Point A

```

tstat  <- qt(p = prob, df = (n-1), lower.tail = TRUE)

lower <- E_pay - sqrt(Var_pay/n)*tstat
upper <- E_pay + sqrt(Var_pay/n)*tstat

return(list(lower, upper))
#Var_pay
}
cat("The following is the expected value of pi")

## The following is the expected value of pi
cat("\n")

beta_point_est(prior_alpha = 2,
               prior_beta  = 10,
               data = algae)

## [1] 0.1608392
cat("The following is the confidence interval of pi")

## The following is the confidence interval of pi
cat("\n")

beta_interval(prior_alpha = 2,
              prior_beta  = 10,
              data = algae,
              prob = .9)

## [[1]]
## [1] 0.1591561
##
## [[2]]
## [1] 0.1625222
cat("\n")

cat("My answers are different when I did the algaeTest, but they made sense to me. I think I am still t

## My answers are different when I did the algaeTest, but they made sense to me. I think I am still thi
cat("\n")

cat("Below, however, I estimate the bayesian credible intervals just for completeness")

## Below, however, I estimate the bayesian credible intervals just for completeness
qbeta(c(0.05,.9), 2 + sum(algae), 10 + length(algae) - sum(algae))

## [1] 0.1265607 0.1891245

c)

beta_low <- function(prior_alpha,
                    prior_beta,
                    data,
                    pi_0) {

```

```

    density <- integrate(function(pay) dbeta(pay, prior_alpha, prior_beta), 0, pi_0)[1]
    return(density)
}
cat("\n")

cat("The following the is probability of (total mass left of) the historical value of pi = 0.2")

## The following the is probability of (total mass left of) the historical value of pi = 0.2
cat("\n")

beta_low(prior_alpha = 2,
         prior_beta = 10,
         data = algae,
         pi_0 = 0.2)

## $value
## [1] 0.6778775

```

d)

The major assumption is the one that leads us to conjugacy. We, as Bayes did, have a closed-form solution for the denominator. But we impose ad-hoc shapes for the distribution of the priors and a known maximum likelihood. How are we sure this is the case in the real world?

e)

I test how the density shifts as we test several values for beta and alpha.

```

library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --
## v ggplot2 3.3.5      v purrr 0.3.4
## v tibble 3.1.6       v dplyr 1.0.7
## v tidyr 1.1.4        v stringr 1.4.0
## v readr 2.1.1       v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag() masks stats::lag()

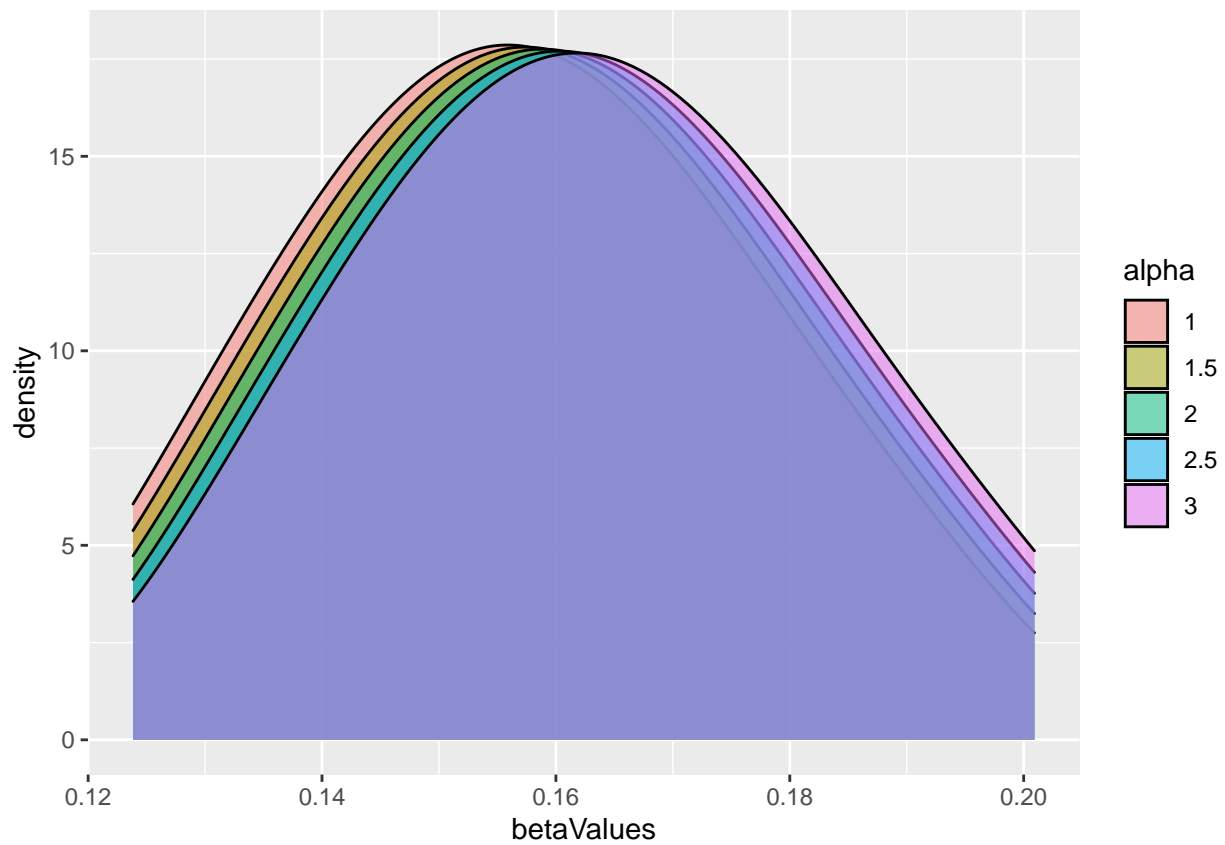
y <- sum(algae)
n <- length(algae)

betaValues <- function(alpha, beta) {
  left <- alpha + y
  right <- beta + n - y
  p <- qbeta(seq(from = 0.05, to = .95, by = 0.05), left, right)
}

beta_tibble <- tibble(alpha = rep(c(1, 1.5, 2, 2.5, 3), each = length(seq(from = 0.05, to = .95, by = 0.05))),
                     betaValues = betaValues(alpha, 10))
beta_tibble$alpha <- as.character(beta_tibble$alpha)

ggplot(beta_tibble, aes(x = betaValues, fill = alpha)) + geom_density(alpha = 0.5)

```



```
alpha_tibble <- tibble(beta = rep(c(8, 8.5, 9, 9.5, 10), each = length(seq(from = 0.05, to = .95, by = 0.05))),
  alpha_tibble <- alpha_tibble %>%
    mutate(betaValues = betaValues(beta, 10))
alpha_tibble$beta <- as.character(alpha_tibble$beta)

ggplot(alpha_tibble, aes(x = betaValues, fill = beta)) + geom_density(alpha = 0.5)
```

