# BDA - Assignment 2

-

## Contents



```
library(aaltobda)
data("algae")
algae_data = algae
algae_n = length(algae_data)
algae_y = sum(algae_data==1)
algae_test <- c(0, 1, 1, 0, 0, 0)
```

## Exercise 1

### a)

**(1)**

$$p(y|\pi) = \binom{n}{y} \cdot \pi^y \cdot (1-\pi)^{n-y}$$

**(2)**

$$p(\pi) = \text{Beta}(\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\pi^{\alpha-1}(1-\pi)^{\beta-1},$$
$$where\ \alpha = 2 \text{ and } \beta = 10$$

**(3)**

$$p(\pi|y) = \frac{p(y|\pi)\cdot p(\pi)}{\int_0^1 p(y|\pi)\cdot p(\pi)} = \text{Beta}(\alpha+y, \beta+(n-y)), = \frac{\Gamma(\alpha+\beta+n)}{\Gamma(\alpha+y)\Gamma(\beta+n-y)}\pi^{\alpha+y-1}(1-\pi)^{\beta+n-y-1},$$
$$where\ \alpha = 2,\ \beta = 10, y = \text{number of positive observations and }, n = \text{number of observations}$$

So the resulting beta function is Beta(46, 240)

### b)

```
beta_point_est = function(prior_alpha = 2, prior_beta = 10, data = algae_test) {
  data_n = length(data)
  data_y = sum(data==1)
  posterior_alpha = prior_alpha + data_y
  posterior_beta = prior_beta + data_n - data_y
  posterior_alpha/(posterior_alpha + posterior_beta)
```

```
}

beta_interval = function(prior_alpha = 2, prior_beta = 10, data = algae_test, prob = 0.9) {
  data_n = length(data)
  data_y = sum(data==1)
  a = prior_alpha + data_y
  b = prior_beta + data_n - data_y
  qbeta(c((1 - prob) / 2, 1 - (1 - prob) / 2), a, b)
}

beta_posterior_mean = beta_point_est(2, 10, algae_data)
beta_posterior_interval = beta_interval(2, 10, algae_data, 0.9)
```

Posterior beta function mean is 0.1608392 and 90% posterior interval is 0.1265607, 0.1978177. The real value is very likely between the 90% posterior interval edges. Thus when going to a new lake in Finland, the probability($\pi$) of finding algae in the water is inside the interval 0.1265607, 0.1978177. Mean is very close to the average of the posterior interval edges, which would imply somewhat identical shape on different sides of the mean.

## c)

```
beta_low = function(prior_alpha = 2, prior_beta = 10, data = algae_test, pi_0 = 0.2) {
  data_n = length(data)
  data_y = sum(data==1)
  a = prior_alpha + data_y
  b = prior_beta + data_n - data_y
  nom = integrate(function(pi) dbeta(pi, a, b),0,pi_0)$val
  denom = integrate(function(pi) dbeta(pi, a, b),0,1)$val
  nom/denom
}

cumulative_probability = beta_low(2, 10, algae_data, 0.2)
```

The probability that $\pi$ is less than or equal to 0.2 is 0.9586136.

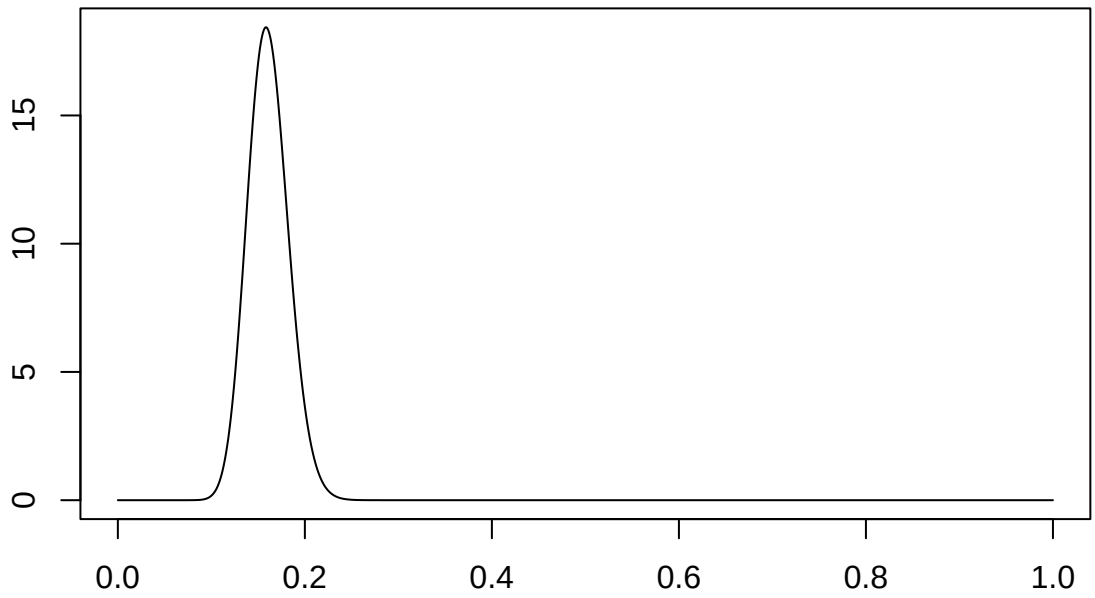
## d)

One of the requirements for the model is that the probability $\pi$ must be independent of other lakes containing algae. In real life algae could for example be spread by animals or small connecting waterways. Prior distribution must be created from real data, or from very sophisticated guess, so that it does not dominate the posterior distribution parameters.

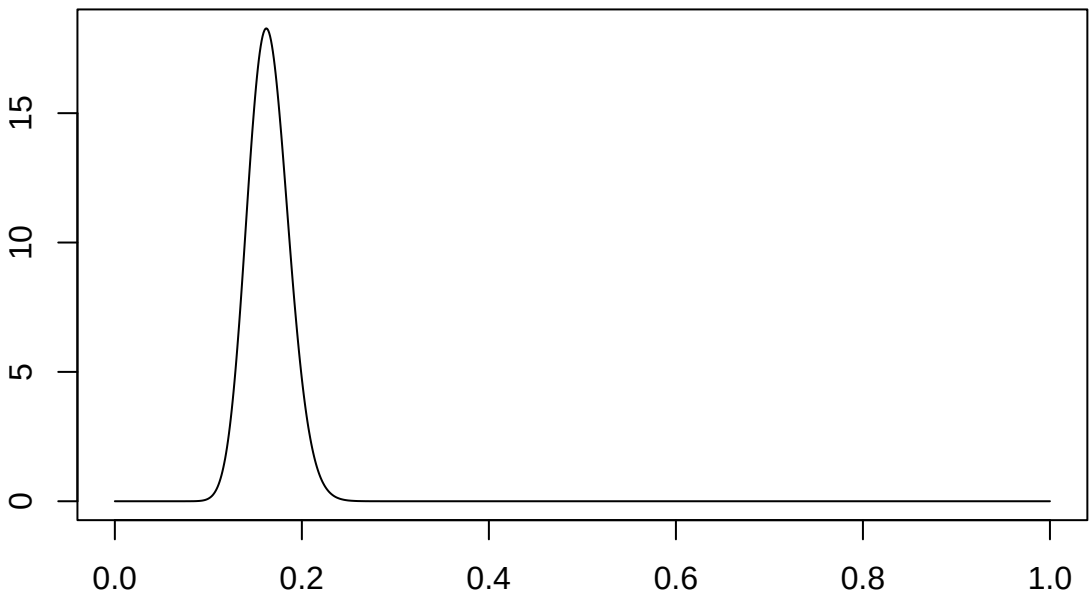
## e)

```
plot_beta = function(prior_alpha, prior_beta, data) {
  data_n = length(data)
  data_y = sum(data==1)
  a = prior_alpha + data_y
  b = prior_beta + data_n - data_y
  sequence = seq(0, 1, 0.001)
  plot(sequence, dbeta(sequence, a, b), xlab = "", ylab = "", type = "l")
}
```

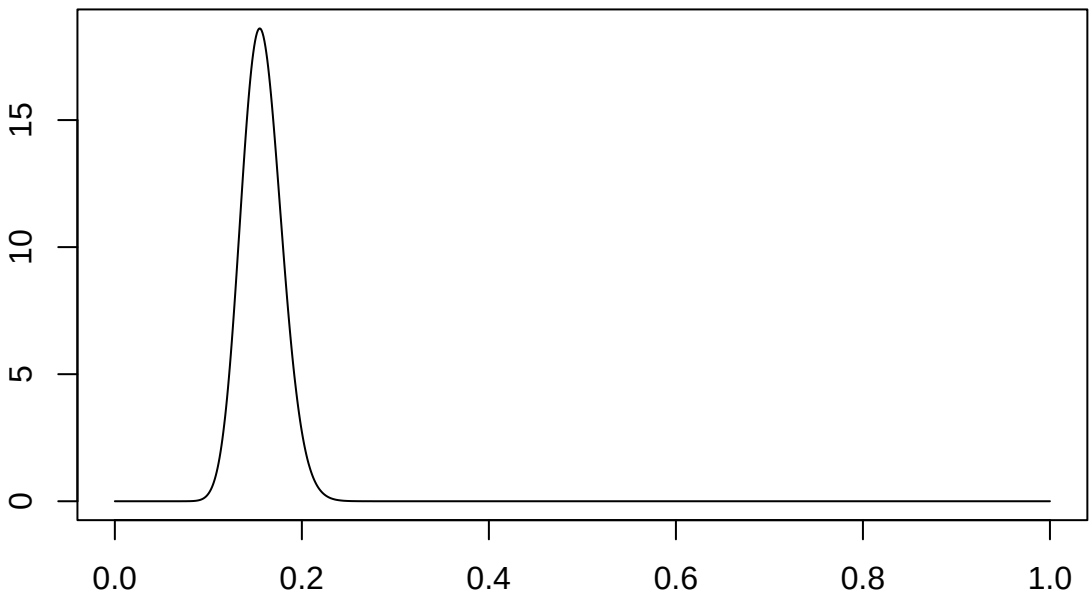
```
plot_beta(2, 10, algae_data)
```



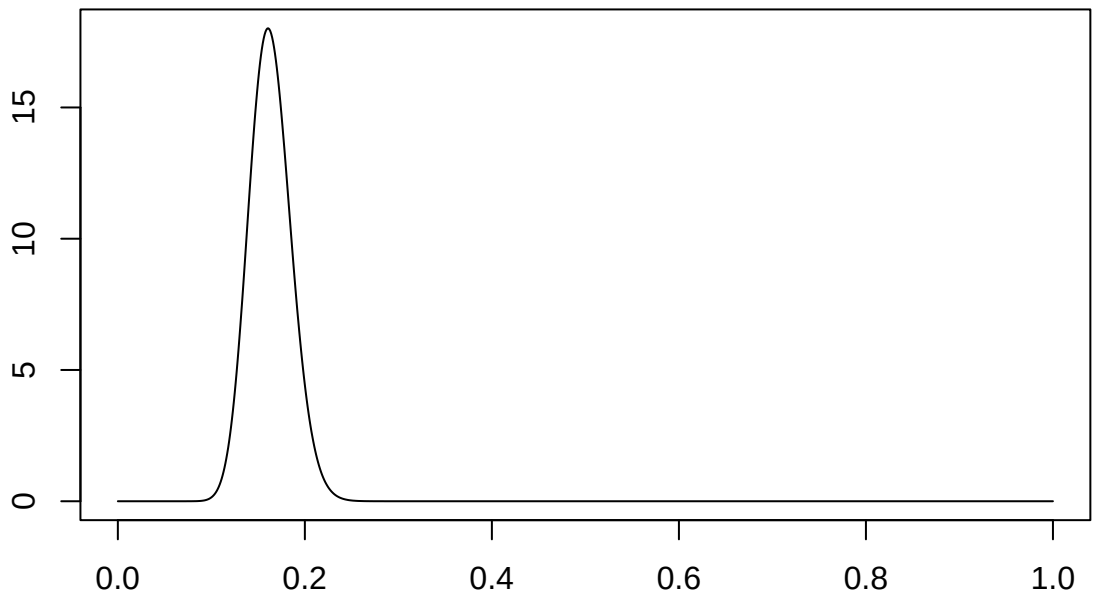

```
plot_beta(3, 9, algae_data)
```



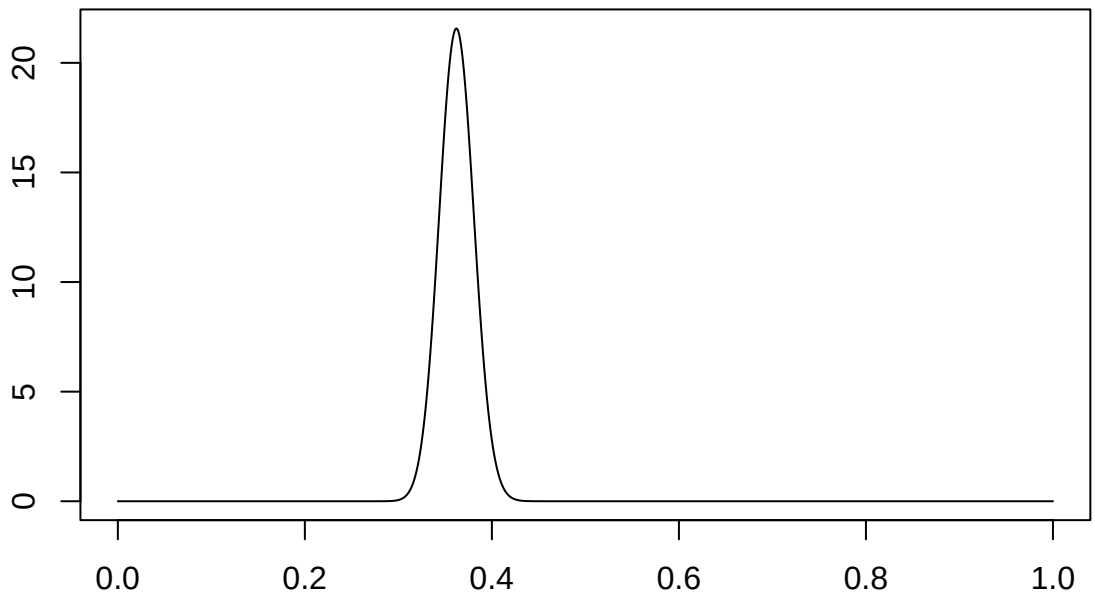

```
plot_beta(1, 11, algae_data)
```




```
plot_beta(1, 1, algae_data)
```




```
plot_beta(201, 201, algae_data)
```




I assume that the Beta prior was calculated from uniform prior with 10 test cases(n = 10) with y being 1. As the prior beta was based on less data than the measured data set, the measured data set basically dominates posterior function parameters. Changing the prior beta function parameters within this 10 test case window will not affect the posterior distribution. If the prior was based on large data set, such as 400 samples as in the last plot, it would pull the posterior more towards its parameters. Using infinite data set would make the plot "very sharp" at the mean value for $\pi$ independently from whatever the prior distribution was. Data set size plays a part here! By parameters I mean alpha and beta of the beta function.

```
library(markmyassignment)
assignment_path <-
paste("https://github.com/avehtari/BDA_course_Aalto/",
"blob/master/assignments/tests/assignment2.yml", sep="")
set_assignment(assignment_path)
```

```
## Assignment set:
## assignment2: Bayesian Data Analysis: Assignment 2
## The assignment contain the following (3) tasks:
## - beta_point_est
## - beta_interval
## - beta_low
```

```
# To check your code/functions, just run
mark_my_assignment()
```

```
## v |  OK F W S | Context
## / |   0       | task-1-subtask-1-tests                                    / |   0       | beta
## / |   0       | task-2-subtask-1-tests                                    / |   0       | beta
## / |   0       | task-3-subtask-1-tests                                    / |   0       | beta
```

```
##
## == Results =======================================================================
## Duration: 0.1 s
##
## [ FAIL 0 | WARN 0 | SKIP 0 | PASS 15 ]
## You're a coding rockstar!
```