Inference for binomial proportion (computer)

a)

prior = Beta (2, 10)
alpha prior = 2
beta_prior = 10

prior_median = $\frac{alfa_{prior}}{alfa_{prior} + beta_{prior}} = \frac{2}{2+10} = 0.1666$

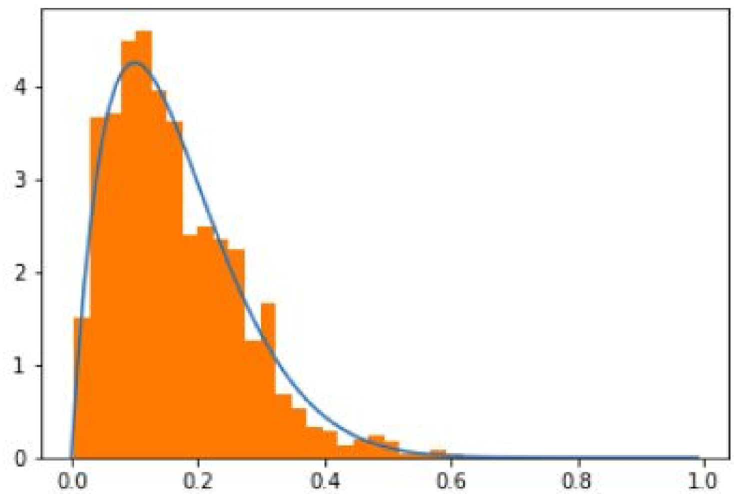
y (data) = Beta (44, 230)
alpha_y = 44
beta_y = 230

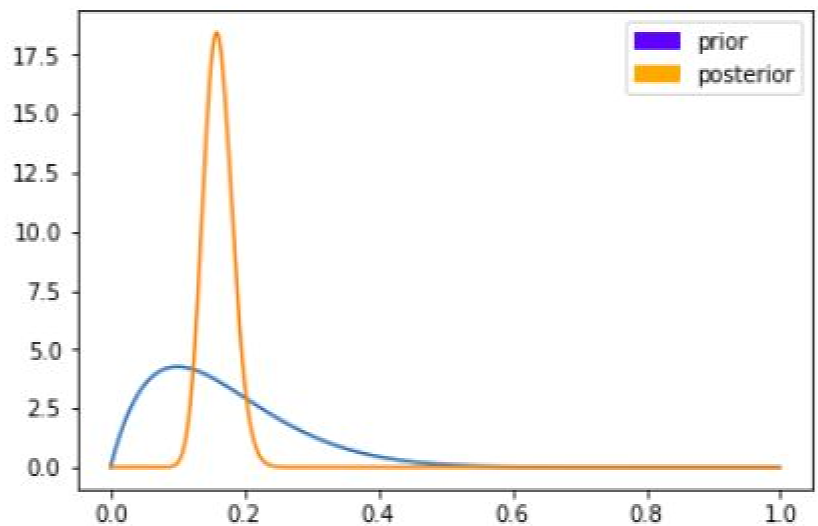
y_median = $\frac{alfa_y}{alfa_y + beta_y} = \frac{44}{44+230} = 0.1605$

posterior = Beta (alpha_prior + alpha_y, beta_prior + beta_y)
alpha_post = 46
beta_post = 240

posterior_median = $\frac{alfa_{post}}{alfa_{post} + beta_{post}} = \frac{46}{286} = 0.1608$

confidence interval 25%: 0.12065
confidence interval 95% : 0.13300





According to the observations and the prior knowledge, we can say that the expected value is similar (the prior and the data are almost identical).

b) The probability that the proportion of the monitoring sites with detectable algae levels π smaller than 0.2 is **0.9586.**

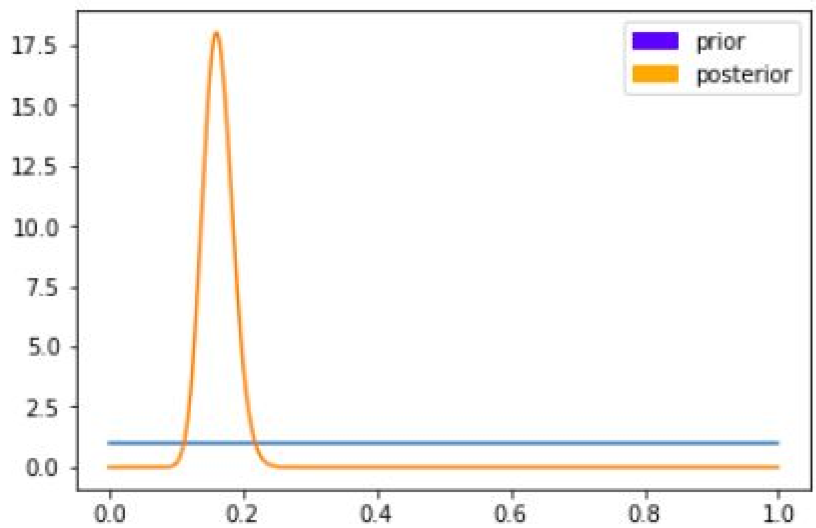
c) We have to assume that for all the lakes, the expected value and the distribution are the same.

d)

- alpha_prior = 1
  beta_prior = 1

  prior_median = 0.5
  posterior_median = 0.1630

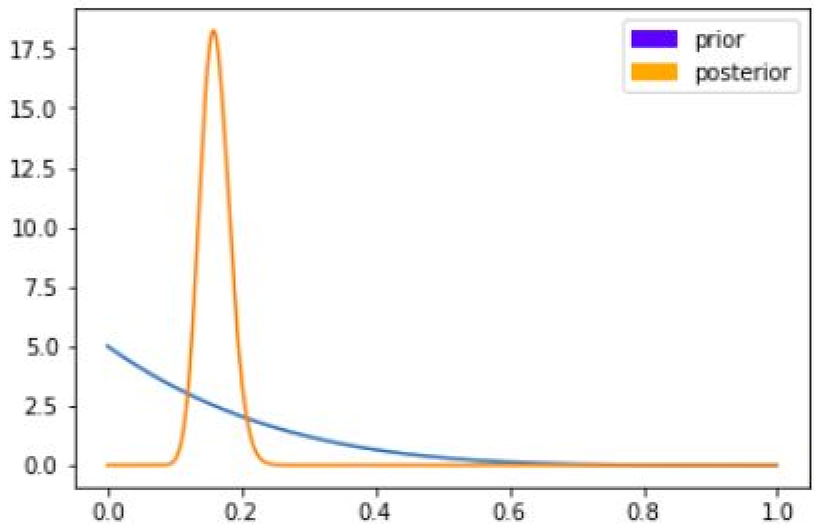
  confidence interval 25%: 0.1219
  confidence interval 95%: 0.1345



- alpha_prior = 1
  beta_prior = 5

  prior_median = 0.1666
  posterior_median = 0.1607

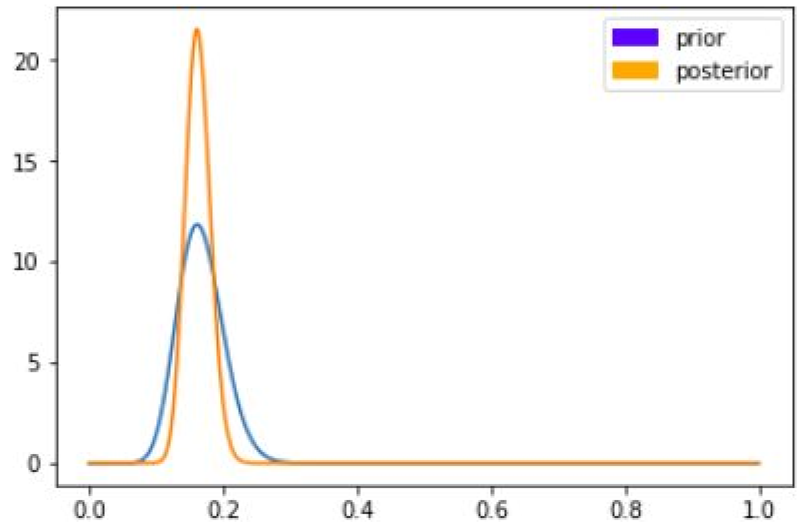
  confidence interval 25%: 0.1201
  confidence interval 95%: 0.1326



- alpha_prior = 20
  beta_prior = 100

  prior_median = 0.1666
  posterior_median = 0.1624

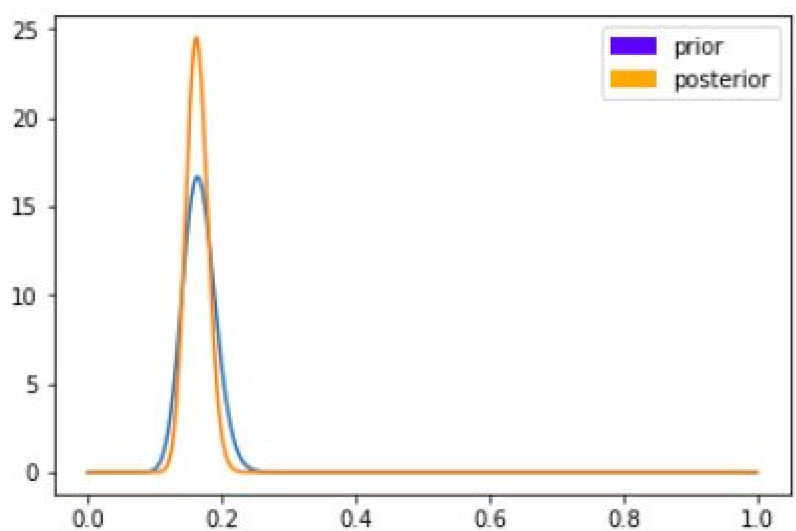
  confidence interval 25%: 0.1277
  confidence interval 95%: 0.1385



- alpha_prior = 40
  beta_prior = 200

  prior_median = 0.1666
  posterior_median = 0.1634

  confidence interval 25%: 0.1327
  confidence interval 95%: 0.1423



From the analysis, we can see that for small values of alpha and beta (which have same mean as Beta(2, 10)), the posterior medians and the confidence intervals are quite similar and they do not change significantly.

**********APPENDIX*************

SOURCE CODE

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import matplotlib.patches as mpatches
from scipy.stats import beta


data = pd.read_csv('algae.txt', sep="\n", header=None)
n = data[0].count()

alfa_prior = 2
beta_prior = 10

alfa_y = 44
beta_y = 230

alfa_post = alfa_y + alfa_prior
beta_post = beta_y + beta_prior

y = alfa_y / (alfa_y + beta_y)
prior = alfa_prior / (alfa_prior + beta_prior)
posterior = alfa_post / (alfa_post + beta_post)

print('posterior mean: ', posterior)
print('confidence interval 25%:', beta.ppf(0.025, alfa_post, beta_post))
print('confidence interval 95%:', beta.ppf(0.095, alfa_post, beta_post))

x = np.arange(0, 1, 0.01)
density = beta.pdf(x, alfa_prior, beta_prior)

sample = beta.rvs(alfa_prior, beta_prior, size = 1000)

plt.plot(x, density)
plt.hist(sample, density=True, bins = 25)
```

```
plt.show()


x = np.arange(0, 1, 0.001)

density = beta.pdf(x, alfa_prior, beta_prior)
density_post = beta.pdf(x, alfa_post, beta_post)

sample = beta.rvs(alfa_post, beta_post, size = 1000)
print(beta.ppf(0.095, alfa_post, beta_post))

blue_patch = mpatches.Patch(color='blue', label='prior')
orange_patch = mpatches.Patch(color='orange', label='posterior')
plt.legend(handles=[blue_patch, orange_patch])

plt.plot(x, density)
plt.plot(x, density_post)

# We can see that the expected value is similar


cumulative = beta.cdf(0.2, alfa_post, beta_post)
print(cumulative)


alfa_prior = 6
beta_prior = 15

alfa_post = alfa_y + alfa_prior
beta_post = beta_y + beta_prior

y = alfa_post / (alfa_post + beta_post)
prior = alfa_prior / (alfa_prior + beta_prior)
posterior = alfa_post / (alfa_post + beta_post)
print('prior: ', prior)
print('posterior: ', posterior)
print('confidence interval 25%:', beta.ppf(0.025, alfa_post, beta_post))
print('confidence interval 95%:', beta.ppf(0.095, alfa_post, beta_post))
```

```
x = np.arange(0, 1, 0.001)

density = beta.pdf(x, alfa_prior, beta_prior)
density_post = beta.pdf(x, alfa_post, beta_post)

blue_patch = mpatches.Patch(color='blue', label='prior')
orange_patch = mpatches.Patch(color='orange', label='posterior')
plt.legend(handles=[blue_patch, orange_patch])
plt.plot(x, density)
plt.plot(x, density_post)
```