Assignment 2

anonymous

1 General information

2 Inference for binomial proportion

Removing old variables, loading the library and the data.

```
rm(list = ls())
library(aaltobda)
data("algae")
```

One is given that $\pi = Beta(2, 10)$ and that y follows a binomial model. We can formulate the likelihood $p(y|\pi)$ as

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

and the prior $p(\pi)$ as

$$p(\pi) = Beta(2, 10).$$

Using this information, we can define the posterior as

$$p(\pi|y) = \binom{n}{y}\pi^y(1-\pi)^{n-y} \cdot p(\pi) = Beta(\pi|2 + y, 10 + n - y).$$

Start by creating a function to calculate the posterior alpha and betas, which will also be useful later.

```
prior_alpha = 2
prior_beta = 10

#algae_test <- c(0, 1, 1, 0, 0, 0)

posterior_distribution = function(data, prior_alpha, prior_beta) {
  n = length(data)
  y = sum(data)
  post_alpha = prior_alpha+y
  post_beta = prior_beta+n-y
  res = list(alpha=post_alpha, beta = post_beta)
  return(res)
```

```
}
res = posterior_distribution(algae, prior_alpha , prior_beta)
alpha = res$alpha
beta = res$beta
cat(paste0("alpha: ", alpha), "\n")
```

```
alpha: 46
```

```
cat(paste0("beta: ", beta), "\n")
```

```
beta: 240
```

The results are:

$$p(\pi|y) = Beta(46, 240)$$

2.1 (b)

```
beta_point_est <- function(prior_alpha, prior_beta, data) {
    res = posterior_distribution(data, prior_alpha,prior_beta)
    alpha = res$alpha
    beta = res$beta
    return(alpha/(alpha+beta))
}

#beta_point_est(prior_alpha = 2, prior_beta = 10, data = algae_test) #Test should be 0.2222222

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

```
[1] 0.1608392
```

The mean was used to compute the point estimate which has a value of 0.161

```
beta_interval <- function(prior_alpha, prior_beta, data, prob=0.9) {
    lower = (1-prob)/2
    upper = prob+lower
    interval = c(lower,upper)
    res = posterior_distribution(data, prior_alpha, prior_beta)
    alpha = res$alpha
    beta = res$beta
    res = qbeta(interval, alpha, beta)
    return(res)
    c(0.0846451, 0.3956414)
}
#beta_interval( prior_alpha = 2, prior_beta = 10, data = algae_test) # Test works as result equals t
beta_interval( prior_alpha = 2, prior_beta = 10, data = algae)
```

```
[1] 0.1265607 0.1978177
```

For the 90% posterior interval is computed a lower bound of 0.127 and a upper bound of 0.198.

2.2 (c)

Keep the below name and format for the function to work with `markmyassignment`:

```
# Useful function: pbeta()

beta_low <- function(prior_alpha, prior_beta, data, pi_0=0.2) {
  res = posterior_distribution(data, prior_alpha, prior_beta)
  alpha = res$alpha
  beta = res$beta
  res = pbeta(pi_0, alpha, beta)
  return(res)
}
#beta_low(prior_alpha = 2, prior_beta = 10, data = algae_test, pi_0 = 0.2) # Test works 0.4511238

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

```
[1] 0.9586136
```

There's approximately a $95,9\%$ chance of being less than 0.2. ## (d)

We've made several key assumptions in our analysis. Firstly, we've assumed that the value of (pi) is continuous. Secondly, we've assumed that all monitoring sites are independent of each other and lastly, the sites exhibit identical behavior. In other words, we've assumed that our data follows the principles of independence and identically distributed (i.i.d.). Furthermore, we've assumed that the algae status in the water is binary, meaning it's either present (1) or absent (0). These are the requirements for us to be able to use binomial- and, by extension, beta-distributions on the data.

2.3 (e)

We will utilize a prior probability of $\pi_0 = 0.2$, as indicated in part c) and derived from historical records, to compute various prior distributions

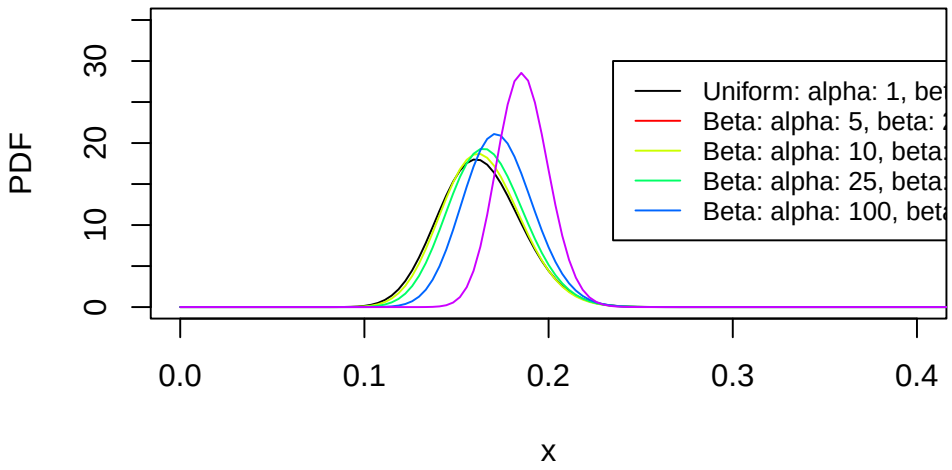
```
E = 0.2
prior_alpha = c(1,5,10,25,100)
prior_beta = c(1,-(prior_alpha[-1]/E)*(E-1))
res = posterior_distribution(algae, prior_alpha, prior_beta)
alpha = res$alpha
beta = res$beta
x= seq(from=0,to=1, by=1/270)

par(mfrow=c(1,1))
plot(x, dbeta(x,alpha[1],beta[1]),
     type='l', col='black',
     ylim=c(0,35),
     xlim=c(0,0.4),
     ylab="PDF")
```

```
legend_text = list()

for (i in 2:5) {
  lines(x, dbeta(x,alpha[i],beta[i]), type='l', col = rainbow(5)[i])
  legend_text[[i-1]] = paste0("Beta: alpha: ",prior_alpha[i], ", beta: ", prior_beta[i])
}

legend(0.235,30, legend=c(paste0("Uniform: alpha: ",prior_alpha[1], ", beta: ", prior_beta[1]), lege
       col=c("black", rainbow(5)), lty=1, cex=0.8)
```



Results are presented in the table bellow. One can see that increasing the prior knowledge makes the beta posterior distribution narrower and shifts towards the prior mean. Furthermore, one can see that the posterior means are mostly within the interval that was computed in the b part. Observing the prior knowledge $\alpha+\beta$, is equal to 500 one can see our earlier calculated posterior mean falls outside the 90% interval.

alpha + beta (Prior)	mean (Prior)	mean (Post)	90% interval lower bound/	upper bound
2	0.5	0.163	0.128	0.201
25	0.2	0.164	0.130	0.200
50	0.2	0.167	0.134	0.202
125	0.2	0.173	0.143	0.205
500	0.2	0.186	0.164	0.209

Chat-GPT 3.5 was used to better formulate the analysis and to brainstorm on different ways to tackle the problems. All answers have been veryfied by the author.