Homework 2

Sara Shao

2021-09-02

1a.

The data can be simulated as follows:

```r
# set a seed
set.seed(123)
# create the observed data
obs.data <- rbinom(n = 100, size = 1, prob = 0.01)
# inspect the observed data
head(obs.data)
```

```
## [1] 0 0 0 0 0 0
```

```r
tail(obs.data)
```

```
## [1] 0 0 0 0 0 0
```

```r
length(obs.data)
```

```
## [1] 100
```

```r
obs.data
```

```
##    [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
##   [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##   [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```
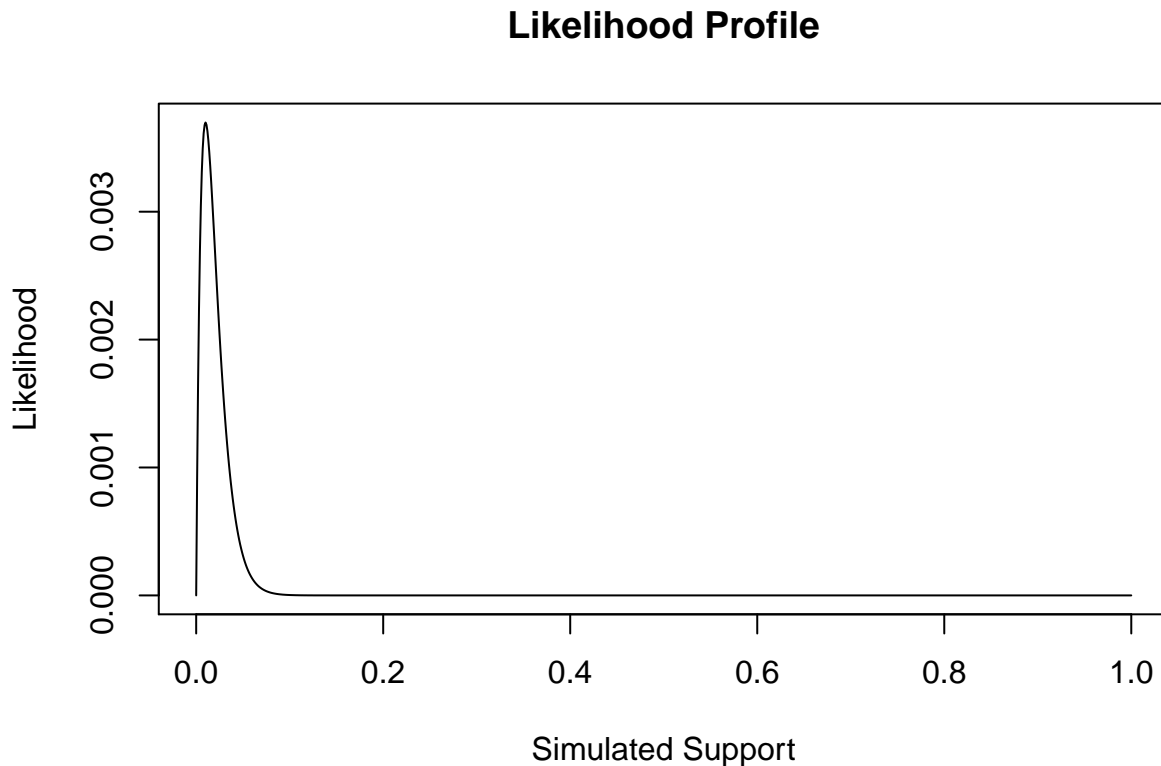
Write a function that takes as its inputs that data you simulated (or any data of the same type) and a sequence of θ values of length 1000 and produces Likelihood values based on the Binomial Likelihood. Plot your sequence and its corresponding Likelihood function.

The likelihood function is given below. Since this is a probability and is only valid over the interval from $[0, 1]$ we generate a sequence over that interval of length 1000.

```r
### Bernoulli LH Function ###
# Input: obs.data, theta
# Output: bernoulli likelihood
myBernLH <- function(obs.data, theta) {
  N <- length(obs.data)
  x <- sum(obs.data)
  LH <- (theta^x)*((1-theta)^(N-x))
  return(LH)
}

### Plot LH for a grid of theta values ###
# Create the grid #
theta.sim <- seq(from = 0, to = 1, length.out = 1000)
# Store the LH values
```

```
sim.LH <- myBernLH(obs.data, theta.sim)
# Create the Plot
plot(theta.sim, sim.LH, type = "l", main = "Likelihood Profile",
     xlab = "Simulated Support", ylab = "Likelihood")
```

Likelihood Profile

1b.

Write a function that takes as its inputs prior parameters `a` and `b` for the Beta-Bernoulli model and the observed data, and produces the posterior parameters you need for the model. **Generate and print** the posterior parameters for a non-informative prior i.e. $(a,b) = (1,1)$ and for an informative case $(a,b) = (3,1)$}.

```
postParam <- function(a, b, obs.data) {
  n <- length(obs.data)
  x <- sum(obs.data)
  post_a <- x + a
  post_b <- n - x + b
  return(c(post_a, post_b))
}
```

Non-informative

```
postParam(1, 1, obs.data)
```

```
## [1]   2 100
```

Informative

```
postParam(3, 1, obs.data)
```

```
## [1]   4 100
```

1c.

Create two plots, one for the informative and one for the non-informative case to show the posterior distribution and superimpose the prior distributions on each along with the likelihood. What do you see? Remember to turn the y-axis ticks off since superimposing may make the scale non-sense.
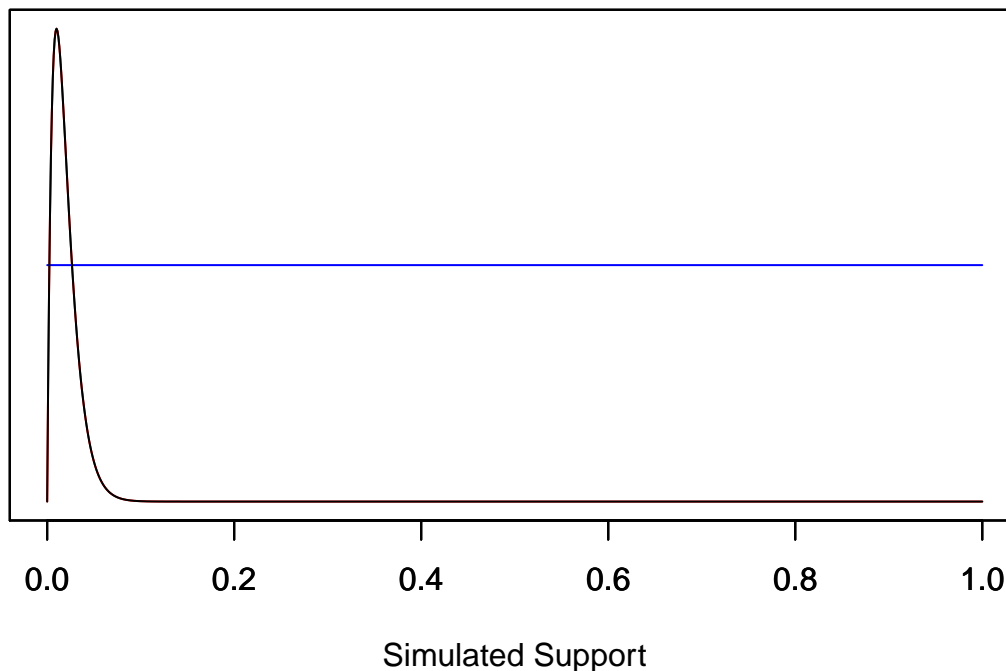
Non-informative

```
theta.sim <- seq(from = 0, to = 1, length.out = 1000)

non_a <- postParam(1, 1, obs.data)[1]
non_b <- postParam(1, 1, obs.data)[2]

sim.LH <- myBernLH(obs.data, theta.sim)

#posterior is red
plot(theta.sim, dbeta(theta.sim, non_a, non_b), type = "l", col = "red",
     yaxt = "n", xlab = "", ylab = "")
par(new = TRUE)
#prior is blue
plot(theta.sim, dbeta(theta.sim, 1, 1), type = "l", col = "blue",
     yaxt = "n", xlab = "", ylab = "")
par(new = TRUE)
#likelihood is black
plot(theta.sim, sim.LH, type = "l",
     yaxt = "n", xlab = "Simulated Support", ylab = "")
```
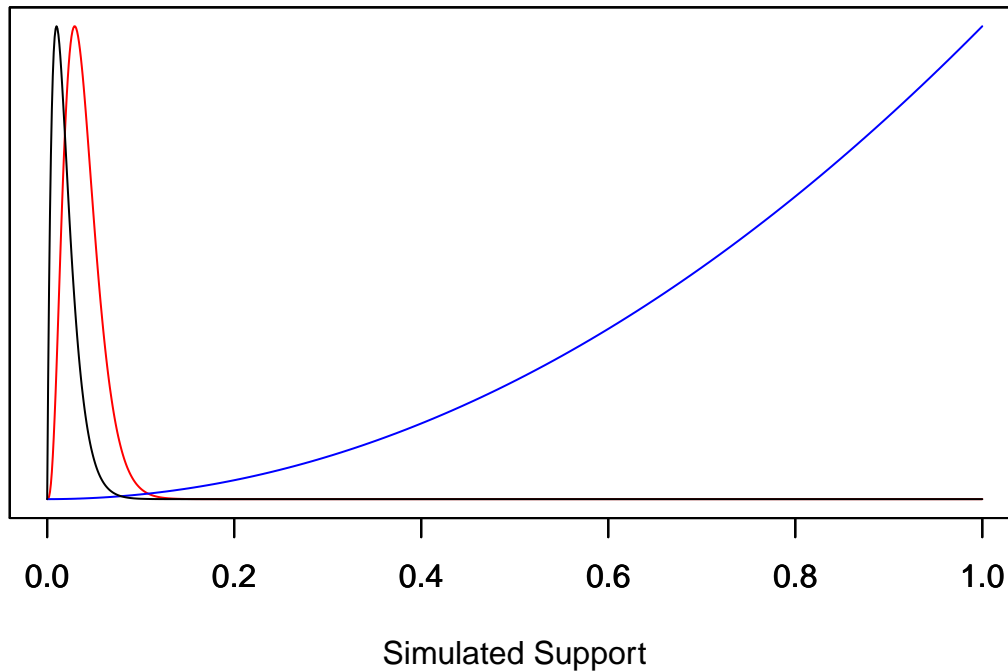

Informative

```
inf_a <- postParam(3, 1, obs.data)[1]
inf_b <- postParam(3, 1, obs.data)[2]

#posterior is red
plot(theta.sim, dbeta(theta.sim, inf_a, inf_b), type = "l", col = "red",
```

```
      yaxt = "n", xlab = "", ylab = "")
par(new = TRUE)
#prior is blue
plot(theta.sim, dbeta(theta.sim, 3, 1), type = "l", col = "blue",
      yaxt = "n", xlab = "", ylab = "")
par(new = TRUE)
#likelihood is black
plot(theta.sim, sim.LH, type = "l",
      yaxt = "n", xlab = "Simulated Support", ylab = "")
```


Simulated Support

In the graph for the non-informative case, the posterior distribution is pretty much exactly proportional to the likelihood distribution. In the informative case, however, we can see that although the posterior distribution is still similar to the likelihood, the mean is shifted a little bit in the the direction of the prior mean.

2c.

```
data <- c(20.9, 69.7, 3.6, 21.8, 21.4, 0.4, 6.7, 10.0)
# sequence of theta values
theta <- seq(0, 1, length.out=100)
x <- sum(data)
n <- length(data)
# calculate prior and posterior distributions
a <- 0.1
b <- 1.0
prior <- dgamma(theta, shape = a, rate = b)
posterior <- dgamma(theta, shape = a + n, rate = b + x)

plot(prior, xlab=expression(theta), ylab="Density", type = "l", col="blue",lty=1, ylim=c(0,30))
lines(posterior, col="red",lty=2)
legend('topright', c("Prior", "Posterior"), lty=c(1,2), col=c("blue","red"))
```

3a.
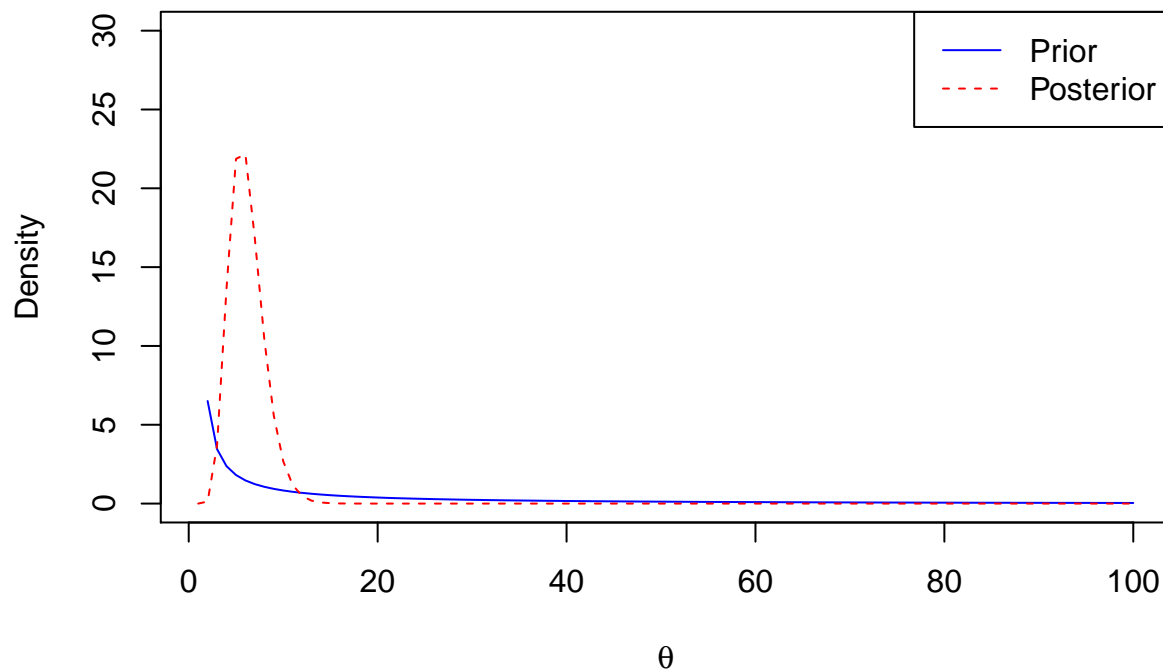
```r
dgalenshore <- function(theta, c, d) {
  result <- (2/gamma(c))*(d^(2*c))*(theta^(2*c-1))*exp(-d^2*theta^2)
  return(result)
}

density1 <- apply(as.matrix(theta.sim), 2, dgalenshore, 3, 4)
density2 <- apply(as.matrix(theta.sim), 2, dgalenshore, 3, 10)
density3 <- apply(as.matrix(theta.sim), 2, dgalenshore, 2, 30)
density4 <- apply(as.matrix(theta.sim), 2, dgalenshore, 15, 5)
```

```r
plot(theta.sim, density1, type = "l", col = "red",
     ylim = c(0,30), ylab = "Density", xlab = expression(theta))
par(new = TRUE)
plot(theta.sim, density2, type = "l", col = "green",
     ylim = c(0,30), ylab = "", xlab = "")
par(new = TRUE)
plot(theta.sim, density3, type = "l", col = "blue",
     ylim = c(0,30), ylab = "", xlab = "")
par(new = TRUE)
plot(theta.sim, density4, type = "l", col = "purple",
     ylim = c(0,30), ylab = "", xlab = "")
```