Assignment #3 STA355H1S

student number: 1003942326

Yulin WANG

2020-03-24

Question1

(a)

By using Bayesian inference, we have the posterior density of (λ, α):

$$\begin{aligned}
\pi(\lambda, \alpha | x_1, ..., x_n) &= \frac{\pi(\lambda, \alpha)\mathcal{L}(\lambda, \alpha)}{\int_0^\infty \int_0^\infty \pi(\lambda, \alpha)\mathcal{L}(\lambda, \alpha) d_\lambda d_\alpha} \qquad \text{for} \quad \lambda, \alpha > 0 \\
&= c(x_1, ..., x_n)\pi(\lambda, \alpha)\mathcal{L}(\lambda, \alpha) \\
&= c(x_1, ..., x_n) \cdot \frac{1}{10000} exp(-\lambda/100) exp(-\alpha/100) \cdot \frac{\lambda^{n\alpha}\{\Pi_{i=1}^n x_i^{\alpha-1}\} exp(-\lambda \sum_{i=1}^n x_i)}{[\Gamma(\alpha)]^n} \\
&= c(x_1, ..., x_n) \cdot \frac{1}{10000} exp(-\frac{\alpha}{100}) \cdot \{\Pi_{i=1}^n x_i^{\alpha-1}\} \cdot \frac{\lambda^{n\alpha}}{[\Gamma(\alpha)]^n} \cdot exp\{-\lambda(\frac{1}{100} + \sum_{i=1}^n x_i)\}
\end{aligned}$$

where $c(x_1, ..., x_n)$ is the normalizing constant depending on the data $x_1, ..., x_n$ and $c(x_1, ..., x_n) = \{\int_0^\infty \int_0^\infty \pi(\lambda, \alpha)\mathcal{L}(\lambda, \alpha) d_\lambda d_\alpha\}^{-1}$

Given the posterior density of (λ, α), we can determine the posterior density of α by integrating over the other parameter λ:

$$\begin{aligned}
\pi(\alpha | x_1, ..., x_n) &= \int_0^\infty \pi(\lambda, \alpha | x_1, ..., x_n) d_\lambda \\
&= c(x_1, ..., x_n) \cdot \frac{1}{10000} exp(-\frac{\alpha}{100}) \cdot \{\Pi_{i=1}^n x_i^{\alpha-1}\} \cdot \frac{1}{[\Gamma(\alpha)]^n} \\
&\cdot \int_0^\infty \lambda^{n\alpha} \cdot exp\{-\lambda(\frac{1}{100} + \sum_{i=1}^n x_i)\} d_\lambda \\
&= c(x_1, ..., x_n) \cdot \frac{1}{10000} exp(-\frac{\alpha}{100}) \cdot exp\{(\alpha - 1)\sum_{i=1}^n ln(x_i)\} \cdot \frac{1}{[\Gamma(\alpha)]^n} \\
&\cdot \Gamma(n\alpha + 1) \cdot (\frac{1}{100} + \sum_{i=1}^n x_i)^{-(n\alpha+1)} \\
&= c(x_1, ..., x_n) \cdot \frac{1}{10000} exp(-\frac{\alpha}{100}) \cdot exp\{\alpha \sum_{i=1}^n ln(x_i)\} \cdot exp\{-\sum_{i=1}^n ln(x_i)\} \\
&\cdot \frac{\Gamma(n\alpha + 1)}{[\Gamma(\alpha)]^n} \cdot (\frac{1}{100} + \sum_{i=1}^n x_i)^{-(n\alpha+1)} \\
&= K(x_1, ..., x_n) \cdot exp\{\alpha \sum_{i=1}^n ln(x_i)\} \cdot exp(-\frac{\alpha}{100}) \cdot \frac{\Gamma(n\alpha + 1)}{[\Gamma(\alpha)]^n} \cdot (\frac{1}{100} + \sum_{i=1}^n x_i)^{-(n\alpha+1)} \\
&= K(x_1, ..., x_n) \frac{\Gamma(n\alpha + 1)}{[\Gamma(\alpha)]^n} exp\{\alpha \sum_{i=1}^n ln(x_i) - \frac{\alpha}{100}\} (\frac{1}{100} + \sum_{i=1}^n x_i)^{-(n\alpha+1)}
\end{aligned}$$

where $K(x_1, ..., x_n) = \frac{1}{10000} exp\{-\sum_{i=1}^n ln(x_i)\} \cdot c(x_1, ..., x_n)$

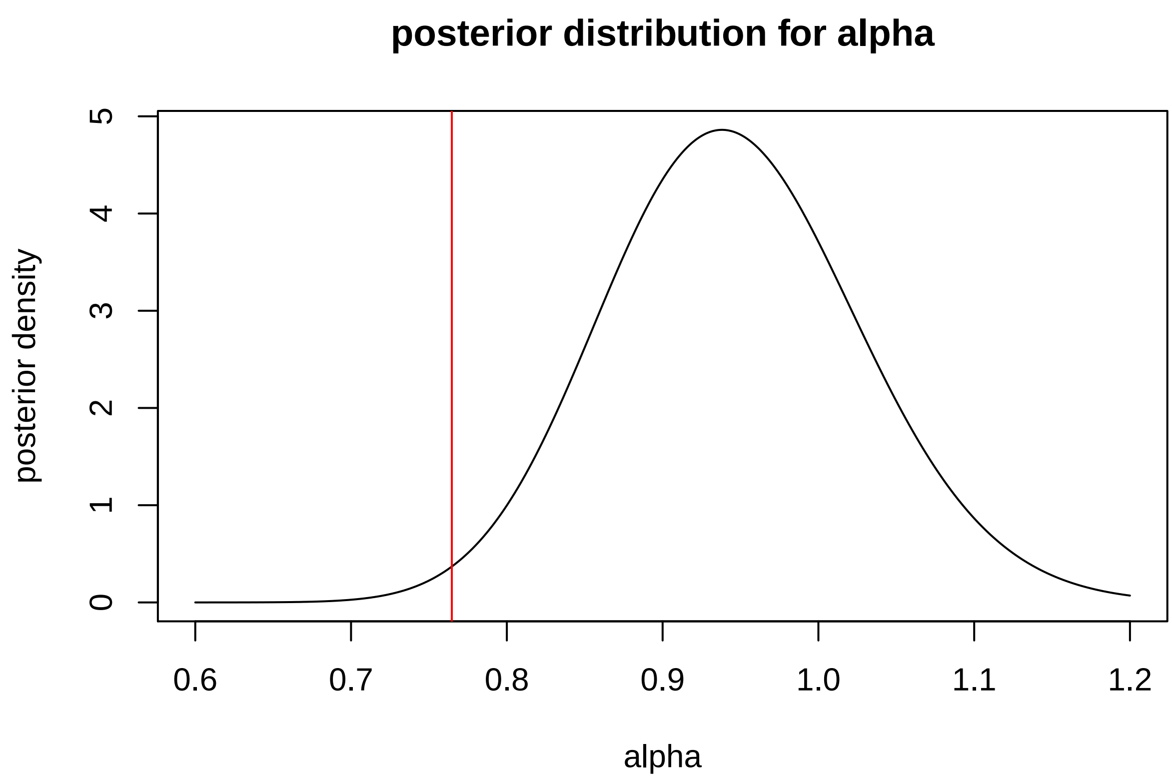
(b)

```
# write a functin to compute pre-normalized posterior
get_prenorm <- function(x, alpha){
  n <- length(x)
  # compute ln(g(alpha))
  lnpost <- lgamma(n*alpha+1) - n*lgamma(alpha) +
    alpha*sum(log(x)) - alpha/100 - (n*alpha+1)*log(1/100+sum(x))
  lnpost <- lnpost - max(lnpost) # subtract maximum
  postunnormed <- exp(lnpost) # pre-normalized posterior
  postunnormed
}

aircon <- scan("aircon.txt") # load the aircon data
alpha_hat <- (mean(aircon)^2)/ var(aircon) # a simple MoM estimate of alpha
alpha_hat
```

```
## [1] 0.7646995
```

```
# implement the trapezoidal rule using h = 1/1000 and N = 600.
alpha <- c(600:1200)/1000 # 601 values of alpha
aircon_prenorm <- get_prenorm(aircon, alpha) # compute pre-normalized posterior
mult <- c(1/2,rep(1,599),1/2) # multipliers for trapezoidal rule
norm <- sum(mult*aircon_prenorm)/1000 # integral evaluated using trapezoidal rule
postnormed <- aircon_prenorm/norm # normalized posterior
plot(alpha, postnormed, type="l", ylab="posterior density",
     main = "posterior distribution for alpha")
abline(v=alpha_hat, col = 'red') # add vertical line for MoM estimated alpha
```

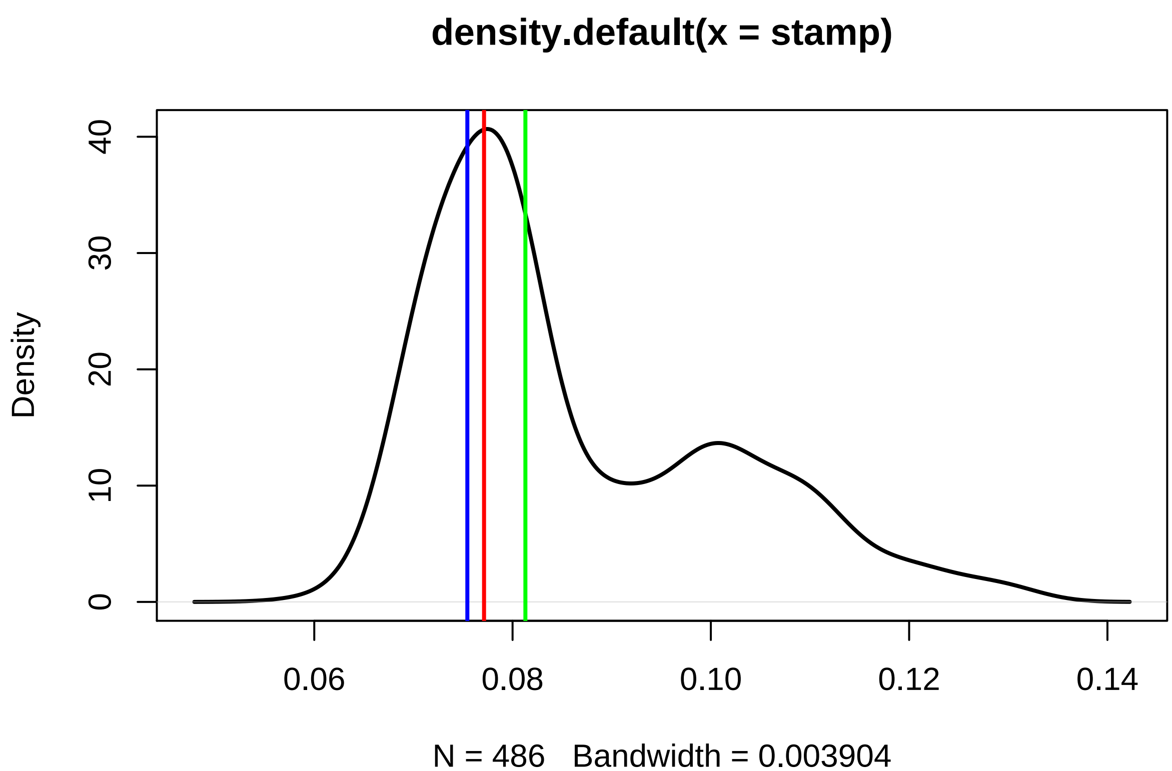


Question2

(a)

```
venter <- function(x, tau=1/2) {
  x <- sort(x)
  n <- length(x)
  m <- ceiling(tau*n)
  x1 <- x[1:(n-m+1)]
  x2 <- x[m:n]
  j <- c(1:(n-m+1))
  len <- x2-x1
  k <- min(j[len==min(len)])
  (x[k]+x[k+m-1])/2
}

stamp <- scan("stamp.txt") # load the stamp data
plot(density(stamp), lwd = 2)
abline(v = venter(stamp), col = 'blue', lwd = 2) # defult value 0.5
abline(v = venter(stamp, 0.629), col = 'red', lwd = 2)
abline(v = venter(stamp, 0.7), col = 'green', lwd = 2)
```



Note that the above density seems to have one clear global maximum around the red vertical line. Thus, τ needs to be about 0.629 in order that the estimate "makes sense".

(b)

```r
# write a function for simulate 1000 times and estimate the MSE of venter estimator
get_mse <- function(tau, n, alpha){
  sim <- NULL
  for (i in 1:1000){
    x <- rgamma(n, alpha)
    sim[i] <- venter(x, tau)
  }
  MSE <- mean((sim-(alpha-1))^2) # MSE of ventor estimator
  MSE
}

# tau=0.5, n=100, alpha=2
get_mse(0.5, 100, 2)
```

```
## [1] 0.1329945
```

```r
# tau=0.5, n=1000, alpha=2
get_mse(0.5, 1000, 2) # best for alpha=2
```

```
## [1] 0.05088367
```

```r
# tau=0.1, n=100, alpha=2
get_mse(0.1, 100, 2)
```

```
## [1] 0.2902677
```

```r
# tau=0.1, n=1000, alpha=2
get_mse(0.1, 1000, 2)
```

```
## [1] 0.0659549
```

```r
# tau=0.5, n=100, alpha=10
get_mse(0.5, 100, 10)
```

```
## [1] 0.7457274
```

```r
# tau=0.5, n=1000, alpha=10
get_mse(0.5, 1000, 10) # best for alpha=10
```

```
## [1] 0.1643127
```

```r
# tau=0.1, n=100, alpha=10
get_mse(0.1, 100, 10)
```

```
## [1] 1.883046
```

```r
# tau=0.1, n=1000, alpha=10
get_mse(0.1, 1000, 10)
```

```
## [1] 0.5258655
```

After the 8 simulations, we estimated the corresponding MSEs of the Venter estimator as above. For both $\alpha = 2$ and $\alpha = 10$, the estimated MSE of the Venter estimator for $\tau = 0.5$ and $n = 1000$ is much smaller than other estimates, while holding α same. Thus, for both $\alpha = 2$ and $\alpha = 10$,the Venter estimator for $\tau = 0.5$ and $n = 1000$ seems to be better(on the basis of MSE).

(c)

Note that each term in the summation is a Gamma density function with shape parameter $k = n\alpha$ and scale parameter $\theta = \frac{\hat{\mu}_i}{T_i}$, that is with rate parameter $\frac{1}{\theta} = \frac{T_i}{\hat{\mu}_i}$, so we can use dgamma function for each term.

```
# implement a function to compute fx
fx <- function(x, n, alpha, tau){
  summation <- NULL
  # simulate 1000 times to compute the summation
  for (i in 1:1000){
    y <- rgamma(n, alpha)
    mu <- venter(y, tau)
    t <- sum(y)
    summation <- c(summation, dgamma(x, shape = n*alpha, rate = t/mu))
    }
  fx <- mean(summation)
  fx
}
# plot the estimated density for x = c(1:500)/100
x <- c(1:500)/100
values <- NULL
for (i in x){
  values <- c(values, fx(i, 100, 2, 0.5)) #n=100,alpha=2,tau=0.5
}
plot(x, values, type = "h", lwd = 1,
     main = "estimated density", ylab = "fxhat")
```