STA355 A3

Ke Deng

2021/3/26

Q1 c) #define functions

```
bee <- scan("bees.txt")

logfunction <- function(x, kappa){
  n = length(x)
  s = sqrt(sum(cos(x))^2 + (sum(sin(x)))^2)

  log <- -n*log(2*pi) - n * log(besselI(kappa, 0)) + kappa*sum(cos(x-mean(x)))
  log
}

prenorm <- function(x, kappa, lambda){
  r <- logfunction(x, kappa)
  r <- r - log(lambda) - log(2*pi) - lambda * kappa
  r <- r - max(r)

  result <- exp(r)
  result
}
```

#khat value

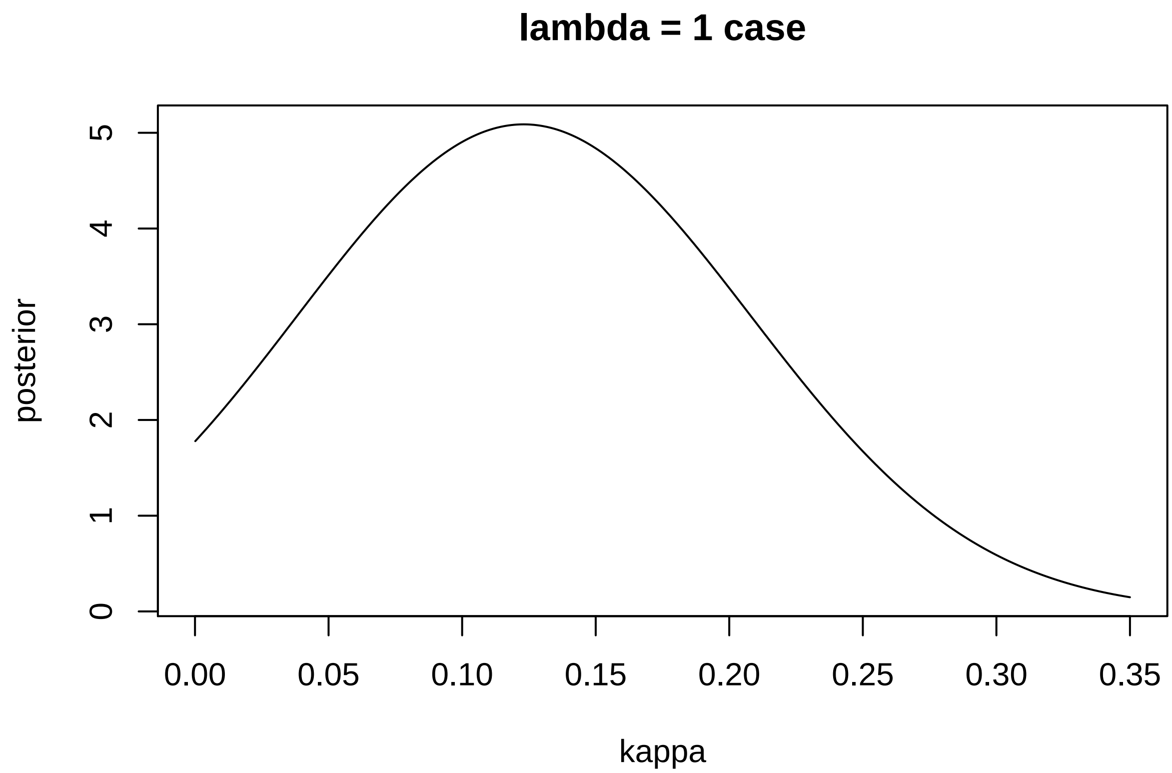
```
x <- bee/180*pi

r <- sqrt(sum(cos(x))^2 + (sum(sin(x)))^2)
n <- length(x)
khat <- (r/n * (2 - r^2/n^2)) / (1 - r^2/n^2)
print(khat)
```

```
## [1] 0.1525488
```

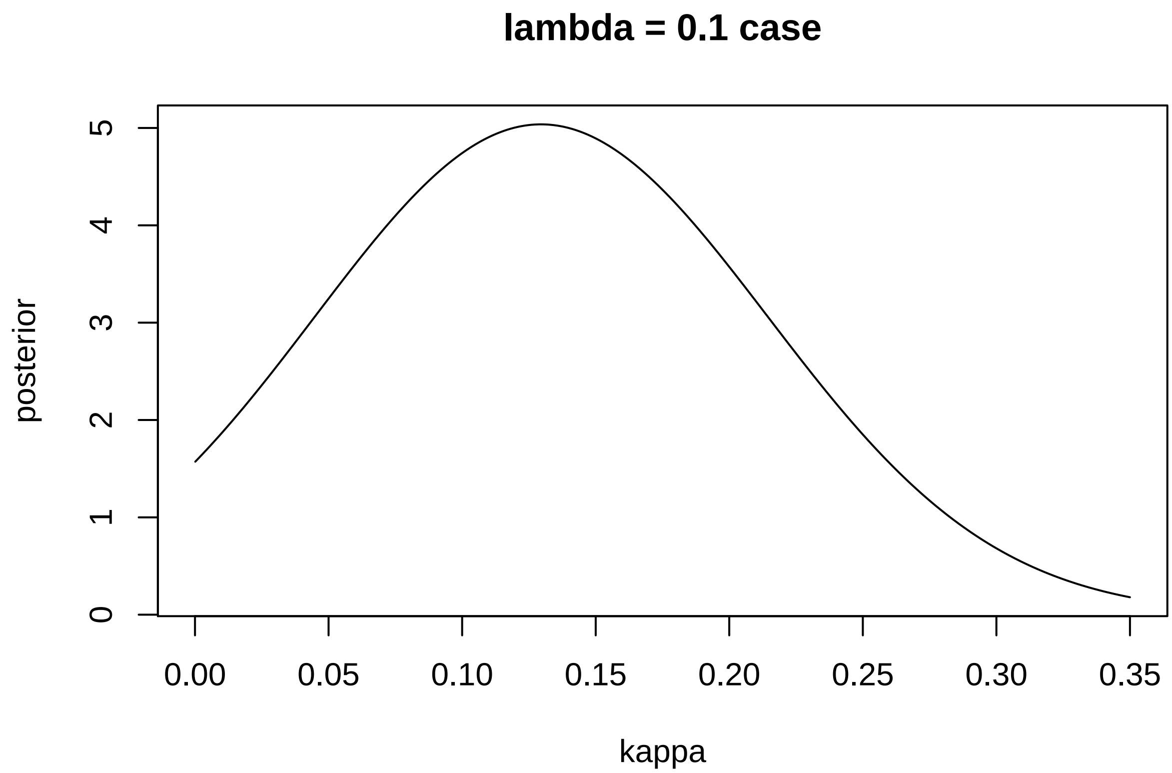
#lambda = 1 case

```
kappa <- c(1:3500)/10000
lambda <- 1
posterior <- prenorm(x,kappa, lambda)
mult <- c(1/2,rep(1, 3499),1/2)
norm <- sum(mult*posterior)/10000
posterior <- posterior/norm
plot(kappa, posterior,type="l",ylab="posterior", main="lambda = 1 case")
```



#lambda = 0.1 case

```
kappa <- c(1:3500)/10000
lambda <- 0.1
posterior <- prenorm(x,kappa, lambda)
mult <- c(1/2,rep(1, 3499),1/2)
norm <- sum(mult*posterior)/10000
posterior <- posterior/norm
plot(kappa, posterior,type="l",ylab="posterior", main="lambda = 0.1 case")
```



From the graph of lambda = 1 and lambda =0.1 above, we can see that both have a max posterior density at around 0.13, which helps eliminate the possibility that k = 0. Also by using the formula, we know that k_hat is approximately 0.15, which also matches the graph we have here.

Q1 d)

```r
probability <- function(theta){
  (theta * (2*pi)^(-n))/(theta*(2*pi)^(-n)+(1-theta)*(norm/exp(513.55)))
}

theta <- 0.1
print(probability(theta))
```

```
## [1] 0.5503128
```

```r
theta <- 0.2
print(probability(theta))
```

```
## [1] 0.7335804
```

```r
theta <- 0.3
print(probability(theta))
```

```
## [1] 0.8251824
```

```r
theta <- 0.4
print(probability(theta))
```

```
## [1] 0.8801334
```

```r
theta <- 0.5
print(probability(theta))
```

```
## [1] 0.9167632
```

```r
theta <- 0.6
print(probability(theta))
```

```
## [1] 0.9429252
```

```r
theta <- 0.7
print(probability(theta))
```

```
## [1] 0.9625456
```

```r
theta <- 0.8
print(probability(theta))
```

```
## [1] 0.9778052
```

```r
theta <- 0.9
print(probability(theta))
```

```
## [1] 0.9900125
```

Q2 c)

```r
#install.package("MASS")
n <- c(50, 100, 500, 1000, 5000)

library(MASS)
a <- function(n){
  rep <- 500
  x <- c(1:n)/n
  beta <- NULL

for (i in 1:rep){
  y = rnorm(n)
  r = lmsreg(y~x)
  beta = c(beta, r$coef[2])
}
  return (var50<- var(beta))
}


var <- NULL
for(i in 1:length(n)){
  var = c(var, a(n[i]))
}

norm_var <- log(var)
n <- log(n)
coef(lm(norm_var~ n))
```

```
## (Intercept)           n
##   2.8814745  -0.6964951
```

Q2 d) Cauchy errors

```
n <- c(50, 100, 500, 1000, 5000)
var_cauchy <- NULL
for(i in 1:length(n)){
var_cauchy = c(var_cauchy, a(n[i]))
}
var <- log(var_cauchy)
n <- log(n)
coef(lm(var~n))
```

```
## (Intercept)           n
##   3.1763865  -0.7371568
```

Using Cauchy errors we see that the value of a is about -0.71, which is close enough to the one we did in part c where a is -0.75. Hence the values are correct.