Homework 3

Yanjie Qi, Jianing (Julia) Chen

Due on November 8, 2020 at 11:59 pm

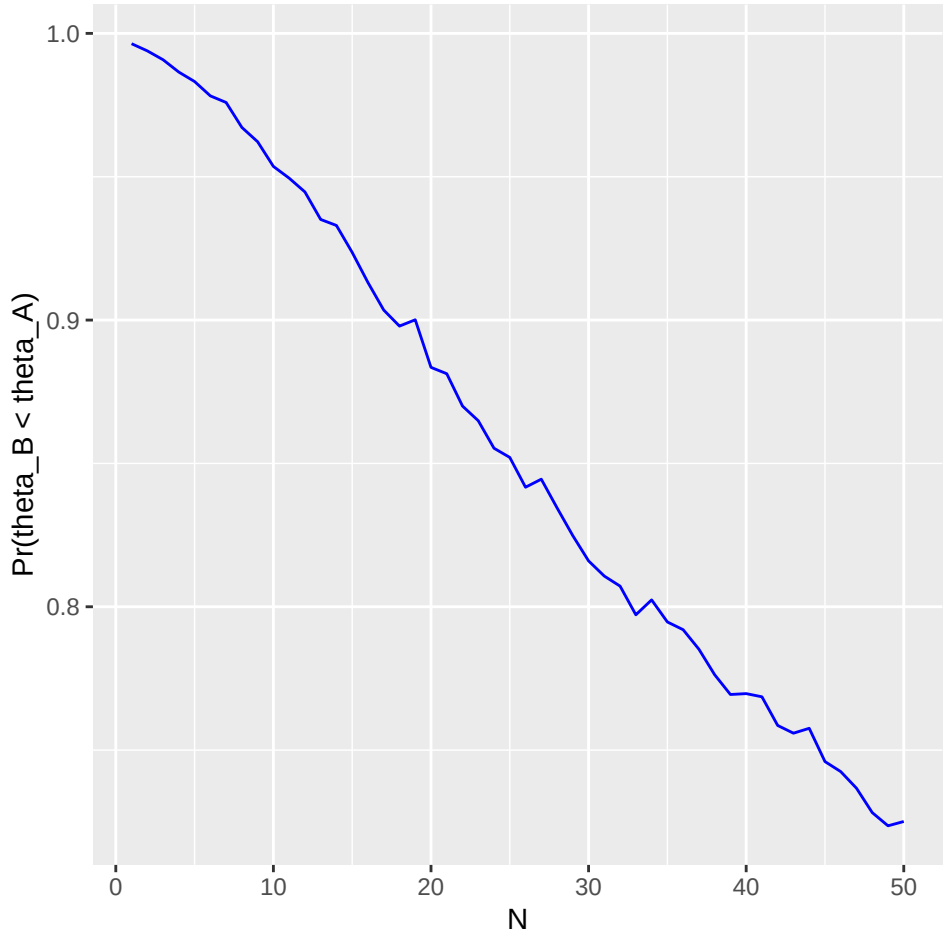
Note: If you are working with a partner, please submit only one homework per group with both names and whether you are taking the course for graduate credit or not. Submit your Rmarkdown (.Rmd) and the compiled pdf on Gauchospace.

Problem 1. Cancer Research in Laboratory Mice

As a reminder from homework 2, a laboratory is estimating the rate of tumorigenesis (the formation of tumors) in two strains of mice, A and B. They have tumor count data for 10 mice in strain A and 13 mice in strain B. Type A mice have been well studied, and information from other laboratories suggests that type A mice have tumor counts that are approximately Poisson-distributed. Tumor count rates for type B mice are unknown, but type B mice are related to type A mice. Assuming a Poisson sampling distribution for each group with rates θ_A and θ_B. We assume $\theta_A \sim$ gamma$(120, 10)$ and $\theta_B \sim$ gamma$(12, 1)$. We observe $y_A = (12, 9, 12, 14, 13, 13, 15, 8, 15, 6)$ and $y_B = (11, 11, 10, 9, 9, 8, 7, 10, 6, 8, 8, 9, 7)$. Now we will actually investigate evidence that Type A mice are have higher rates of tumor formation than Type B mice.

a. For $n_0 \in \{1, 2, ..., 50\}$, obtain $Pr(\theta_B < \theta_A \mid y_A, y_B)$ via Monte Carlo sampling for $\theta_A \sim$ gamma$(120, 10)$ and $\theta_B \sim$ gamma$(12 \times n_0, n_0)$. Make a line plot of $Pr(\theta_B < \theta_A \mid y_A, y_B)$ vs n_0. Describe how sensitive the conclusions about the event $\{\theta_B < \theta_A\}$ are to the prior distribution on θ_B.

```
y_A <- c(12, 9, 12, 14, 13, 13, 15, 8, 15, 6)
y_B <- c(11, 11, 10, 9, 9, 8, 7, 10, 6, 8, 8, 9, 7)

n_0 = 1:50
alpha_A = 120
beta_A = 10
#alpha_B = 12*n_0+113
#beta_B = n_0+13

sumY_A = sum(y_A)
nY_A = length(y_A)
sumY_B = sum(y_B)
nY_B = length(y_B)

prop <- c()
for(n in n_0){
  thetaA= rgamma(10000,alpha_A+sumY_A, beta_A+nY_A)
  thetaB= rgamma(10000,12*n+sumY_B, n+nY_B)
  prop = c(prop, mean(thetaB<thetaA))
}

prob = data.frame("Prob" = prop,"N" = n_0)


ggplot(prob, aes(x = N, y = Prob)) + geom_line(color = 'blue') + ylab("Pr(theta_B < theta_A)")
```

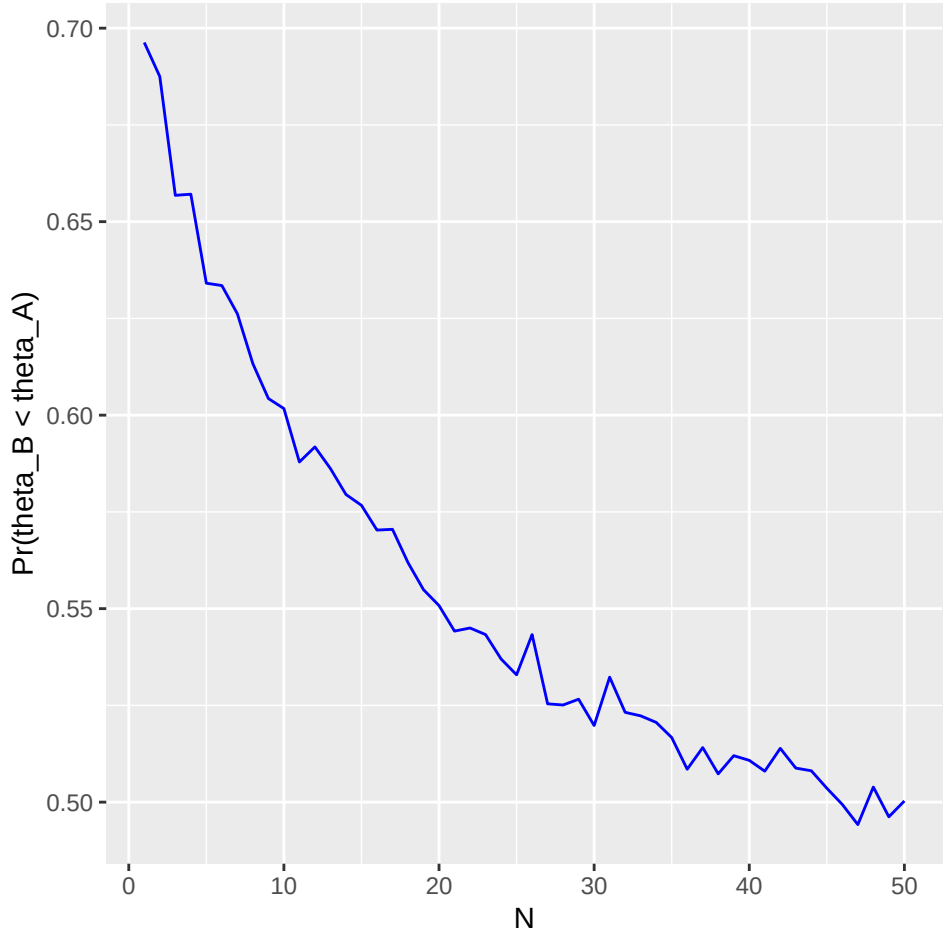


From the plot, it is clear that posterior distribution is sensitive and depends on the prior distribution, the probability for the $\theta_B < \theta_A$ decreases as n0 increases.

b. Repeat the previous part replacing the event $\{\theta_B < \theta_A\}$ with the event $\{\tilde{Y}_B < \tilde{Y}_A\}$, where $\tilde{Y}_A$ and $\tilde{Y}_B$ are samples from the posterior predictive distribution.

```
y_A = c(12, 9, 12, 14, 13, 13, 15, 8, 15, 6)
y_B = c(11, 11, 10, 9, 9, 8, 7, 10, 6, 8, 8, 9, 7)

n_0 <- 1:50
alpha_A = 120
beta_A = 10
alpha_B = 113
beta_B = 13

sumY_A = sum(y_A)
nY_A = length(y_A)
sumY_B = sum(y_B)
nY_B = length(y_B)

prop2 <- c()
for(n in n_0){
  thetaA.mc = rgamma(10000,alpha_A+sumY_A, beta_A+nY_A)
  thetaB.mc= rgamma(10000,12*n+sumY_B, n+nY_B)
```

```
  # Monte Carlo sampling
  yA.mc = rpois(10000,thetaA.mc)
  yB.mc = rpois(10000,thetaB.mc)
  prop2 <- c(prop2, mean(yB.mc<yA.mc))
}


prob2 = data.frame("Prob" = prop2,"N" = n_0)


ggplot(prob2, aes(x = N, y = Prob)) + geom_line(color = 'blue') + ylab("Pr(theta_B < theta_A)")
```



From the plot, it is clear that posterior distribution is sensitive and depends on the prior distribution, the probability for the $\{\tilde{Y}_B < \tilde{Y}_A\}$ also decreases as n0 increases.

c. In the context of this problem, describe the meaning of the events $\{\theta_B < \theta_A\}$ and $\{\tilde{Y}_B < \tilde{Y}_A\}$. How are they different?

In this problem, $\{\theta_B < \theta_A\}$ means the probability of which count rate of type B mice is less than count rate of mice A. $\{\tilde{Y}_B < \tilde{Y}_A\}$ is designed to check the possiblity of which type B mice tumor count is less than type A mice tumor count. The first is to check the possiblity of count rate, namely the prior rate of each group, difference; and the second is to check the difference between two mice groups of actual posterial predictive count getting tumult.

2. Posterior Predictive Model Checking

Model checking and refinement is an essential part of Bayesian data analysis. Let's investigate the adequacy of the Poisson model for the tumor count data. Consider strain A mice only for now, and generate posterior predictive datasets $y_A^{(1)}, ..., y_A^{(1000)}$. Each $y_A^{(s)}$ is a sample of size $n_A = 10$ from the Poisson distribution with parameter $\theta_A^{(s)}$, $\theta_A^{(s)}$ is itself a sample from the posterior distribution $p(\theta_A \mid y_A)$ and y_A is the observed data. For each s, let $t^{(s)}$ be the sample average divided by the sample variance of $y_A^{(s)}$.

a. If the Poisson model was a reasonable one, what would a "typical" value $t^{(s)}$ be? Why?

```
y_A = c(12, 9, 12, 14, 13, 13, 15, 8, 15, 6)
```

The "typical" value of $t^{(s)}$ should be 1 because the expectation value of possion distribution is exactly the same as the variance of it, which should be $\frac{\lambda}{\lambda}$

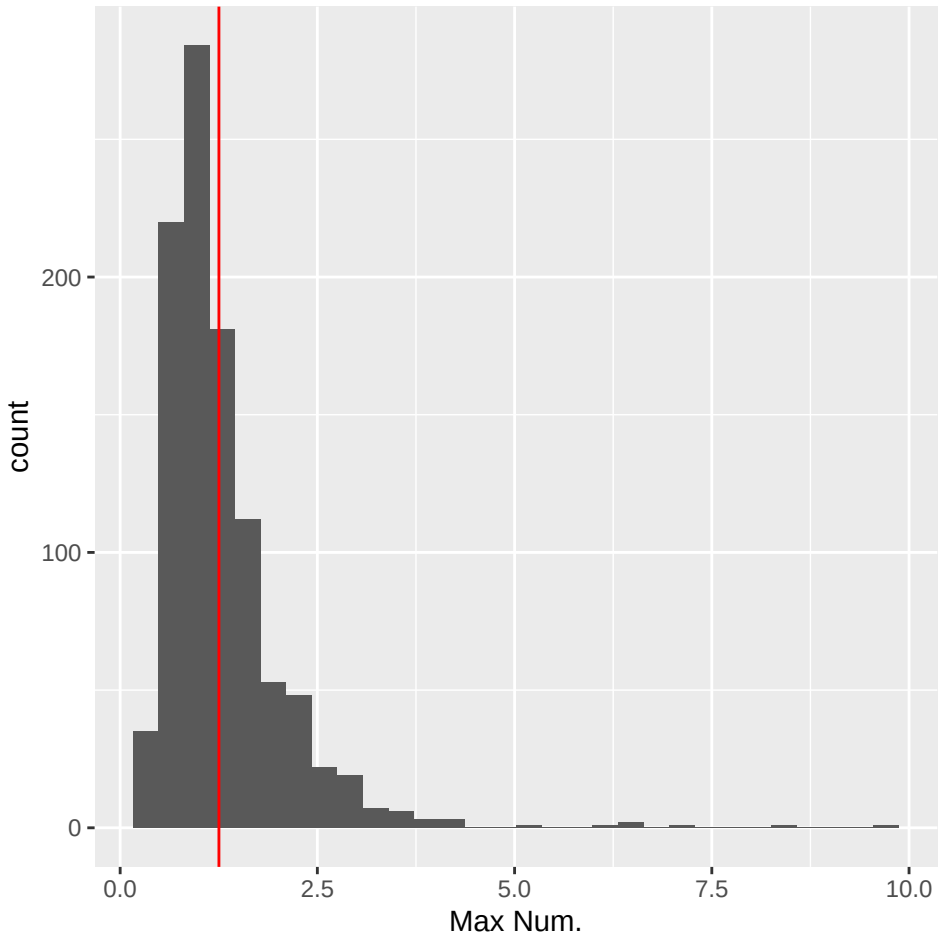
b. In any given experiment, the realized value of t^s will not be exactly the "typical value" due to sampling variability. Make a histogram of $t^{(s)}$ and compare to the observed value of this statistic, $\frac{\text{mean}(y_A)}{\text{var}(y_A)}$. Based on this statistic, make a comment on if the Poisson model seems reasonable for these data (at least by this one metric).

```
set.seed(123)

nsim <- 1000
test_stat_rep <- rep(NA, nsim)

for(i in 1:1000){
  p_post <- rgamma(1, alpha_A+sumY_A, beta_A+nY_A)
  yA_rep <- rpois(nY_A, p_post)
  test_stat <- mean(yA_rep)/var(yA_rep)
  test_stat_rep[i] <- test_stat
}

test_stat
```

```
## [1] 0.6190476
```

```
ggplot(tibble(test_stat_rep), aes(test_stat_rep)) +
  geom_histogram() + xlab("Max Num.") +
  geom_vline(xintercept = mean(y_A)/var(y_A), colour = "red")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

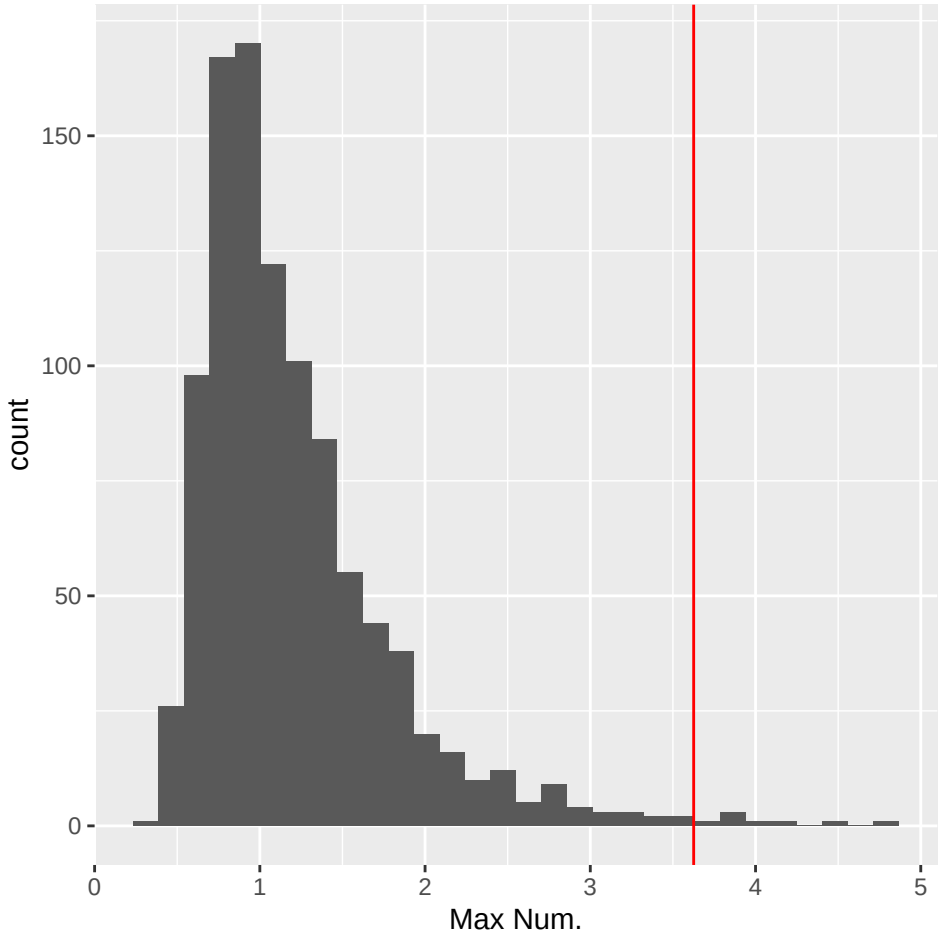


The Poisson model should be reasonable, since the distribution of the histogram follows the Poisson distribution and the observed statistics is within the middle part of the distribution which is around 1.

c. Repeat the part b) above for strain B mice, using Y_B and $n_B = 13$ to generate the samples. Assume the prior distribution $p(\theta_B) \sim \text{Gamma}(12, 1)$. Again make a comment on the Poisson model fit.

```
set.seed(123)
y_B = c(11, 11, 10, 9, 9, 8, 7, 10, 6, 8, 8, 9, 7)

nsim <- 1000
test_stat_rep <- rep(NA, nsim)


for(i in 1:1000){
  p_post <- rgamma(1, alpha_B+sumY_B, beta_B+nY_B)
  yB_rep <- rpois(nY_B, p_post)
  test_stat <- mean(yB_rep)/var(yB_rep)
  test_stat_rep[i] <- test_stat
}

ggplot(tibble(test_stat_rep), aes(test_stat_rep)) +
  geom_histogram() + xlab("Max Num.") +
  geom_vline(xintercept = mean(y_B)/var(y_B), colour = "red")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The Poisson model does not reasonable in this case. Even though the distribution of the histogram follows the Poisson distribution, but the observed statistics is around 3.6 which is very far from the center.

3. Interval estimation with rejection sampling.

a. Use rejection sampling to sample from the following density:

$$p(x) = \frac{1}{4}|\sin(x)| \times I\{x \in [0, 2\pi]\}$$

Use a proposal density which is uniform from 0 to 2π and generate at least 1000 true samples from p(x). Compute and report the Monte Carlo estimate of the upper and lower bound for the 50% quantile interval using the `quantile` function on your samples. Compare this to the 50% HPD region calculated on the samples. What are the bounds on the HPD region? Report the length of the quantile interval and the total length of the HPD region. What explains the difference? Hint: to compute the HPD use the `hdi` function from the `HDInterval` package. As the first argument pass in `density(samples)`, where `samples` is the name of your vector of true samples from the density. Set the `allowSplit` argument to true and use the `credMass` argument to set the total probability mass in the HPD region to 50%.

b. Plot $p(x)$ using the `curve` function (base plotting) or `stat_function` (ggplot). Add lines corresponding to the intervals / probability regions computed in the previous part to your plot using them `segments` function. To ensure that the lines don't overlap visually, for the HPD region set `y0` and `y1` to 0 and for the quantile interval set `y0` and `y1` to 0.01. Make the segments for HPD region and the segment for

quantile interval different colors. Report the length of the quantile interval and the total length of the HPD region, verifying that indeed the HPD region is smaller.

```r
# Rejection sampling and interval construction
library(HDInterval)

density_ratio <- function(x){
  0.25*abs(sin(x))*(ifelse(x >=0 | x <= 2*pi, 1, 0)) / (dunif(x, 0, 2*pi))
  }
M <- optimize(density_ratio, lower = 0, upper = 2*pi, maximum = TRUE)$objective

set.seed(2020)
n <- 1e6
uniform_sample <- runif(n, 0, 2*pi)
accept <- runif(n) < density_ratio(uniform_sample)/M
samples <- uniform_sample[accept]


# hd_region is the result of calling hdi function
hd_region <- hdi(density(samples), allowSplit = TRUE, credMass = 0.5)
print(hd_region)
```

```
##         begin      end
## [1,] 1.050274 2.092318
## [2,] 4.189937 5.231980
## attr(,"credMass")
## [1] 0.5
## attr(,"height")
## [1] 0.2153797
```

```r
print(sprintf("Total region length: %.02f", sum(hd_region[, "end"] - hd_region[,"begin"])))
```

```
## [1] "Total region length: 2.08"
```

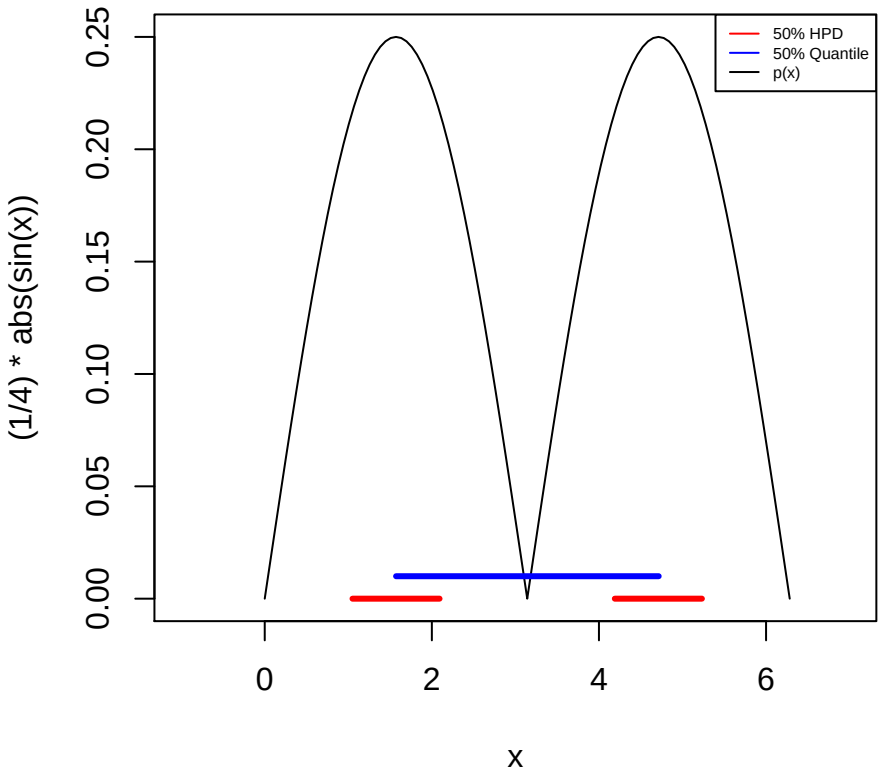
```r
quantile_interval <- quantile(samples,c(0.25,0.75))
print(quantile_interval)
```

```
##      25%      75%
## 1.571691 4.714971
```

```r
print(sprintf("Total region length: %.02f", quantile_interval[2] - quantile_interval[1]))
```

```
## [1] "Total region length: 3.14"
```

```r
## Make the plot

curve((1/4)*abs(sin(x)), from=0, to=2*pi, xlim=c(-1, 7))
# interval segments for HPD region
segments(x0=hd_region[1, 1], y0=0, x1=hd_region[1, 2], y1=0, col="red", lwd=3)
segments(x0=hd_region[2, 1], y0=0, x1=hd_region[2, 2], y1=0, col="red", lwd=3)
# interval line for quantile interval
segments(x0=quantile_interval[1], y0=0.01, x1=quantile_interval[2], y1=0.01, col="blue", lwd=3)
legend('topright', legend=c("50% HPD", "50% Quantile","p(x)"),
       col=c('red','blue','black'),lty=1,cex=0.5)
```



The upper and lower bound for the 50% quantile interval is 1.567067, and 4.711751 respectively. Besides, the 50% HPD region is from (1.033090, 2.153023) to (4.254815, 5.313381). The total length of the HPD region is 2.08 while the length of the quantile interval is 3.14.

The difference is that the HPD region have a higher posterior density than points outside the region, that is because we compute both 50% quantile interval and 50% HPD region, then the points in the HPD region will be more concentrated within the region, and it will have shorter length compared to the length of quantile interval.

In part (b) we plot the segments for HPD region and the segment for quantile interval, and observed that the blue line is longer than red line. Therefore, we can verify that total length of HPD region is shorter compared to the length of quantile interval.