Homework 2

PSTAT 115, Spring 2021

Due on May 9, 2021 at 11:59 pm

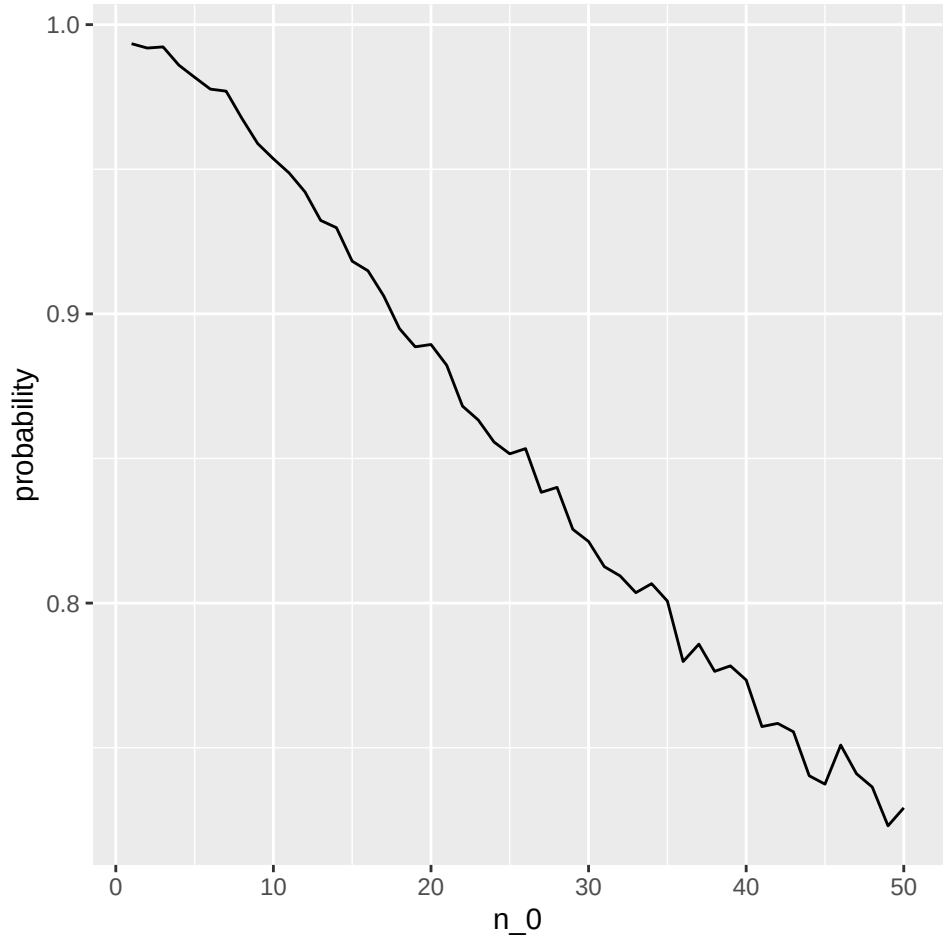
Note: If you are working with a partner, please submit only one homework per group with both names and whether you are taking the course for graduate credit or not. Submit your Rmarkdown (.Rmd) and the compiled pdf on Gauchospace.

Problem 1. Cancer Research in Laboratory Mice

As a reminder from homework 1, 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.

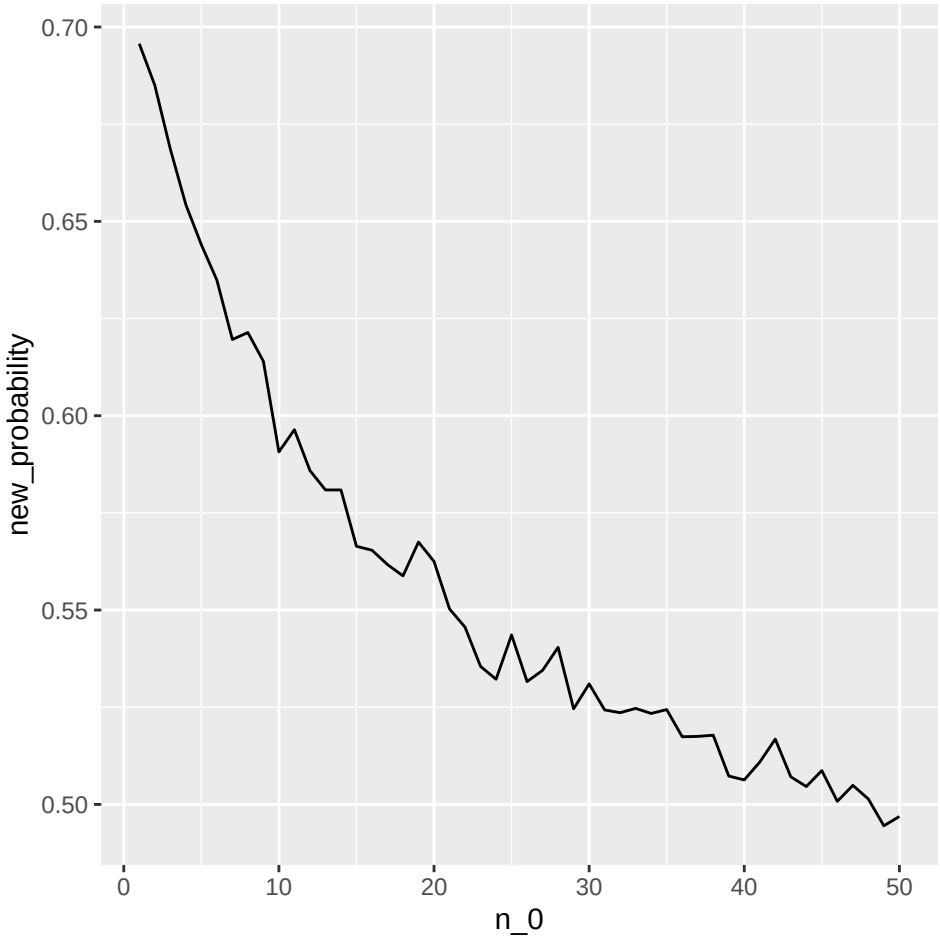
```
y_A <- c(12, 9, 12, 14, 13, 13, 15, 8, 15, 6)
y_B <- c(11, 11, 10, 9, 9, 8, 7, 10, 6, 8, 8, 9, 7)

### BEGIN SOLUTION
n_0 <- 1:50
probability <- c()

for (n in n_0){
theta_a <- rgamma(10000,120 + 117, 10+10)
theta_b <- rgamma(10000,(12 * n) + 113, n + 13)
probability[n] <- mean(theta_b < theta_a)
}

ggplot(data=tibble(n_0,probability),aes(x=n_0,y=probability))+geom_line()
```



From the graph, I would say that the results are fairly sensitive because we can see that the probability of $\{\theta_B < \theta_A\}$ decreases as the prior distribution for θ_B increases by n_0.

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)

### BEGIN SOLUTION
n_0 <- 1:50
new_probability <- c()
for (n in n_0){
  theta_a <- rgamma(10000,120 + 117, 10+10)
  theta_b <- rgamma(10000,(12 * n) + 113, n + 13)
  ytilde_a <- rpois(10000,theta_a)
  ytilde_b <- rpois(10000,theta_b)
  new_probability[n] <- mean(ytilde_b < ytilde_a)
}

ggplot(data=tibble(n_0,y =new_probability),aes(x = n_0, y= new_probability)) + geom_line()
```



Once again, from the graph we can see that the probability of $\{\tilde{Y}_B < \tilde{Y}_A\}$ decreases as the prior distribution for θ_B increases by n_0.

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?

Theta is the parameter of a Poisson, so we can think of it like the expectation of the tumor count. On the other hand, $\{\tilde{Y}_B$ is the realization of the tumor count. Therefore, in the context of this problem $\{\theta_B < \theta_A\}$ is the rate of tumors in group B being less than the rate of tumors for group A. On the other hand, $\{\tilde{Y}_B < \tilde{Y}_A\}$ means the samples from the posterior predictive distribution of tumor count in type B is less than the tumor count in type A.

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 $y_A^{(s)}$.

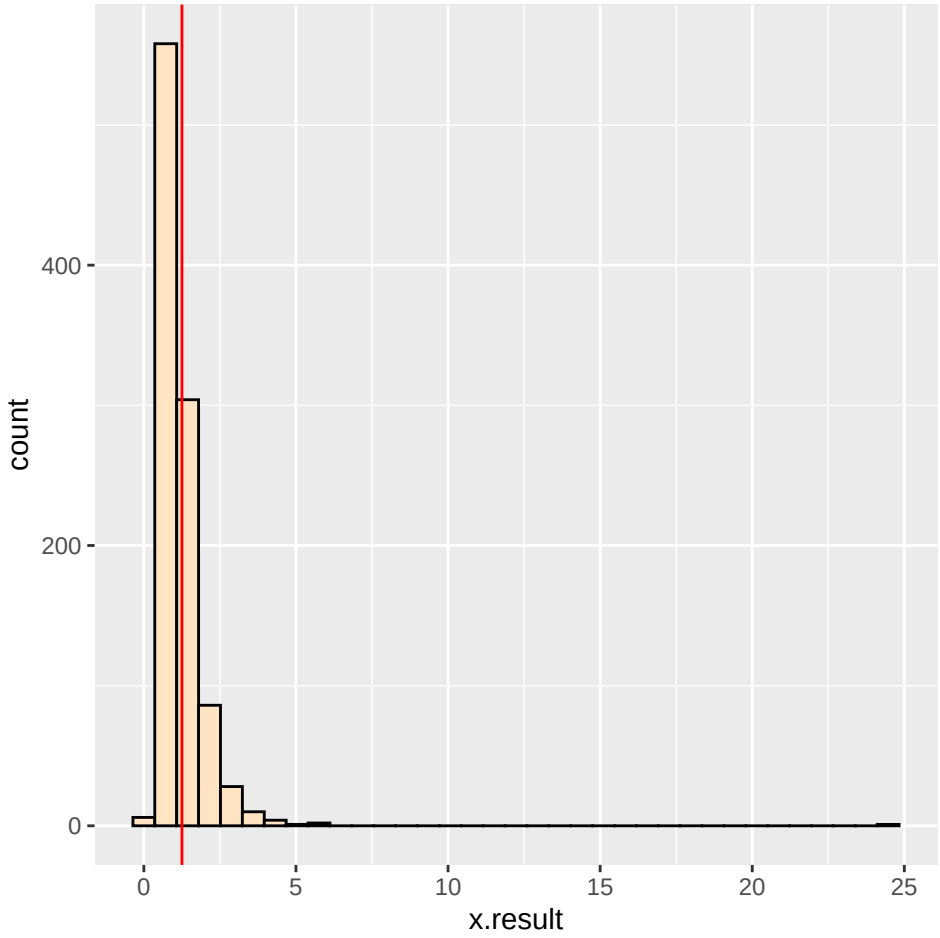
a. If the Poisson model was a reasonable one, what would a "typical" value $t^{(s)}$ be? Why?

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

### BEGIN SOLUTION
typical_val <- mean(y_A)/var(y_A)
```

A typical value for the mean/variance would be 1 in general since for a poisson model the mean and variance are the same value. In our case we have 1.25, which is pretty close, so we would want the observed value to be close to this typical value if the Poisson model was a good fit.

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).

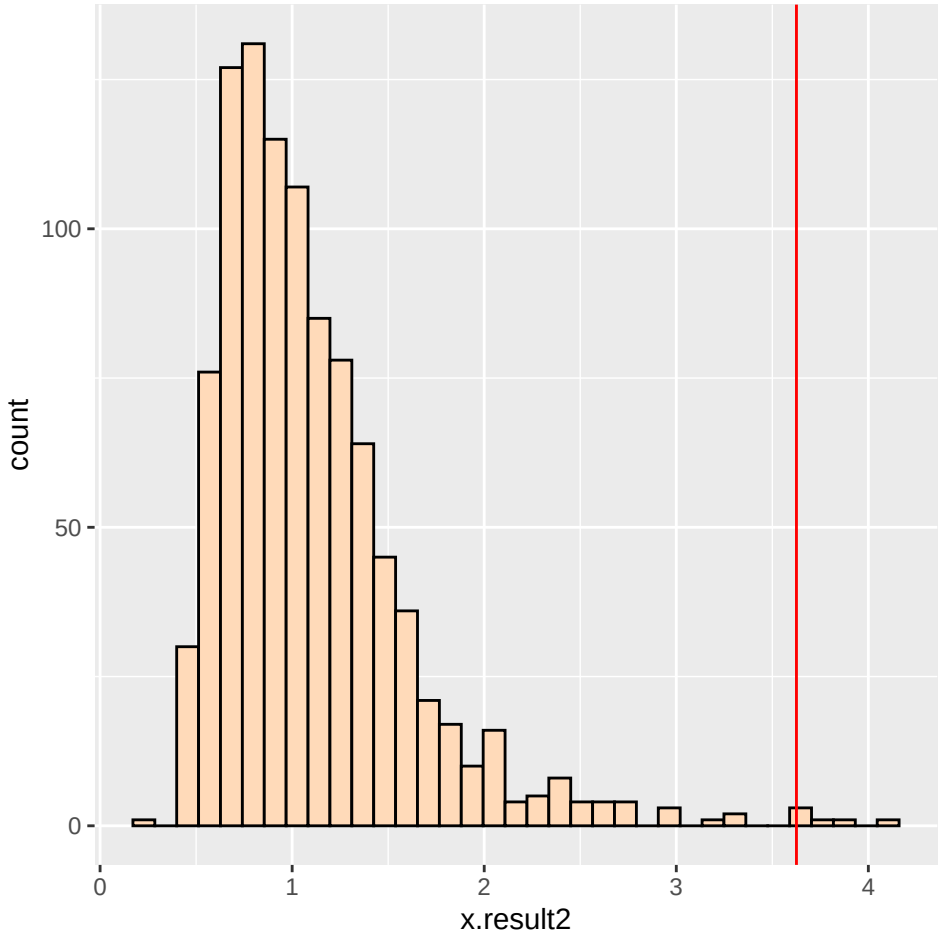
```
### BEGIN SOLUTION
x <- 1000
x.result <- numeric(x)

for(x in 1:x){
  theta_x <- rgamma(10000,120 + 117, 10+10)
  ytilde_x <- rpois(10,theta_x)
  x.result[x] <- mean(ytilde_x)/var(ytilde_x)
}
ggplot(tibble(x.result),aes(x = x.result)) +
  geom_histogram(color='black',fill='bisque',bins = 35)+
  geom_vline(xintercept = typical_val, color='red')
```



From the plot, we can see that the Poisson model seems reasonable for this data because the simulated value is similar to the observed value of the statistic as shown by the red line.

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.

```
y_B = c(11, 11, 10, 9, 9, 8, 7, 10, 6, 8, 8, 9, 7)

### BEGIN SOLUTION
typical_val_B <- mean(y_B)/var(y_B)
x2 <- 1000
x.result2 <- numeric(x2)

for(x in 1:x2){
  theta_x2 <- rgamma(10000,12+ 113, 1 + 13)
  ytilde_x2 <- rpois(13,theta_x2)
  x.result2[x] <- mean(ytilde_x2)/var(ytilde_x2)
}
ggplot(tibble(x.result2),aes(x = x.result2)) +
  geom_histogram(color='black',fill='peachpuff',bins = 35)+
  geom_vline(xintercept = typical_val_B, color='red')
```



From the plot, we can see that the sample is not best suited by the Poisson model. This is because the observed value is not centered around the "typical" value that we would expect if the Poisson model was a good fit.

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%.

```r
#help from classmate Rayne Frantez
#install.packages('HDInterval')
library(HDInterval)
p <- function(x) {
  0.25 * abs(sin(x))/dunif(x, 0, 2 * pi)
}

q <- function(x) {
  dunif(x, 0, 2 * pi)
}

density.ratio <- function(x) {
  p(x) / q(x)
}

M <- optimize(density.ratio, lower = 0, upper = 2*pi, maximum = TRUE)$objective
sample <- runif(1000, 0, 2*pi)
accept <- runif(1000) < density.ratio(sample) / M
samples <- sample[accept]
quantile.interval <- quantile(samples, c(0.25, 0.75))
quantile.interval
```

```
##      25%      75%
## 1.482781 4.730604
```

```r
hd.region <- hdi(density(samples), allowSplit = TRUE, credMass = 0.5)
hd.region
```

```
##          begin      end
## [1,] 0.8733394 2.158187
## [2,] 4.2714236 5.302683
## attr(,"credMass")
## [1] 0.5
## attr(,"height")
## [1] 0.1898057
```

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

```
## [1] "Total region length: 3.25"
```

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

```
## [1] "Total region length: 2.32"
```

The difference between the intervals could be explained by the fact that we are using the vector of true samples from the density when calculating the HPD, while we use just the sample values when calcuating the quantile intervals.

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.

```
### Rejection sampling and interval construction
### BEGIN SOLUTION

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

# SOLUTION
print(hd_region)
```

```
##          begin      end
## [1,] 0.8733394 2.158187
## [2,] 4.2714236 5.302683
## attr(,"credMass")
## [1] 0.5
## attr(,"height")
## [1] 0.1898057
```

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

```
## [1] "Total region length: 2.32"
```

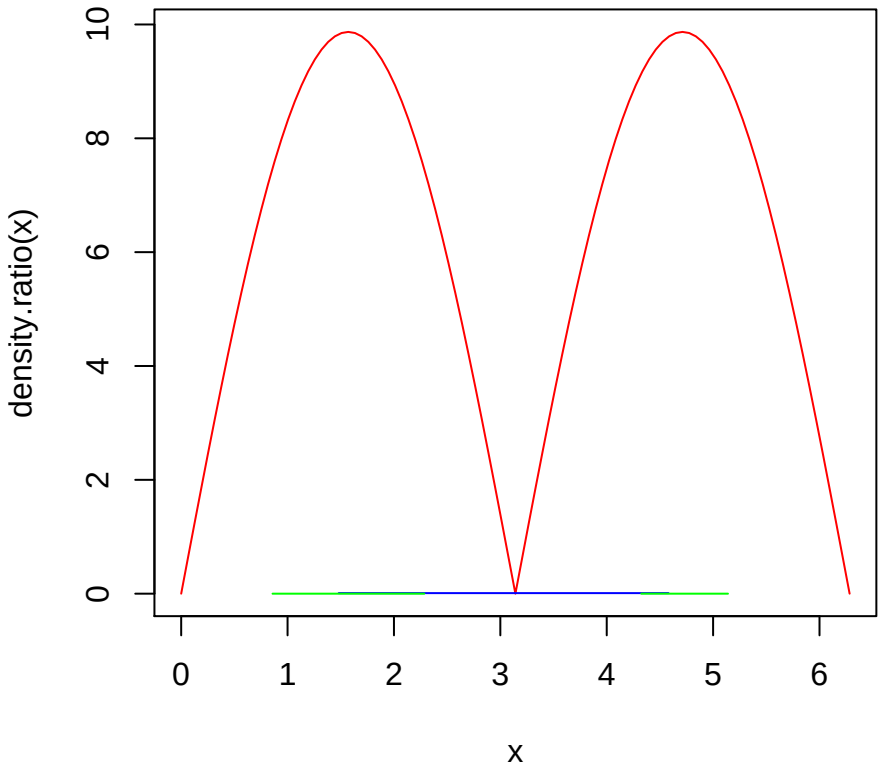
```
quantile_interval <- quantile(samples, c(0.25, 0.75)) # SOLUTION
print(quantile_interval)
```

```
##      25%      75%
## 1.482781 4.730604
```

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

```
## [1] "Total region length: 3.25"
```

```
### Make the plot
### BEGIN SOLUTION
curve(density.ratio,from = 0, to = 2*pi, col = 'red')
segments(1.480691,.01,4.580316,.01, col = 'blue', lwd = 1)
segments(.858682,0,2.285709,0, col = 'green', lwd=1) +
  segments(4.324319,0,5.139763,0, col = 'green',lwd = 1)
```




```
## integer(0)
```

The blue line represents the HPD, the red lines represents p(x) and the green lines represent the quantile.