Homework3a

yi Chen

9/20/2018

1. Exercise 2.21 of BDA

```
# read the data
library('foreign')
library(dplyr)
library(ggplot2)

result <- read.csv('2008ElectionResult.csv')
data <- read.dta('pew_research_center_june_elect_wknd_data.dta')
```

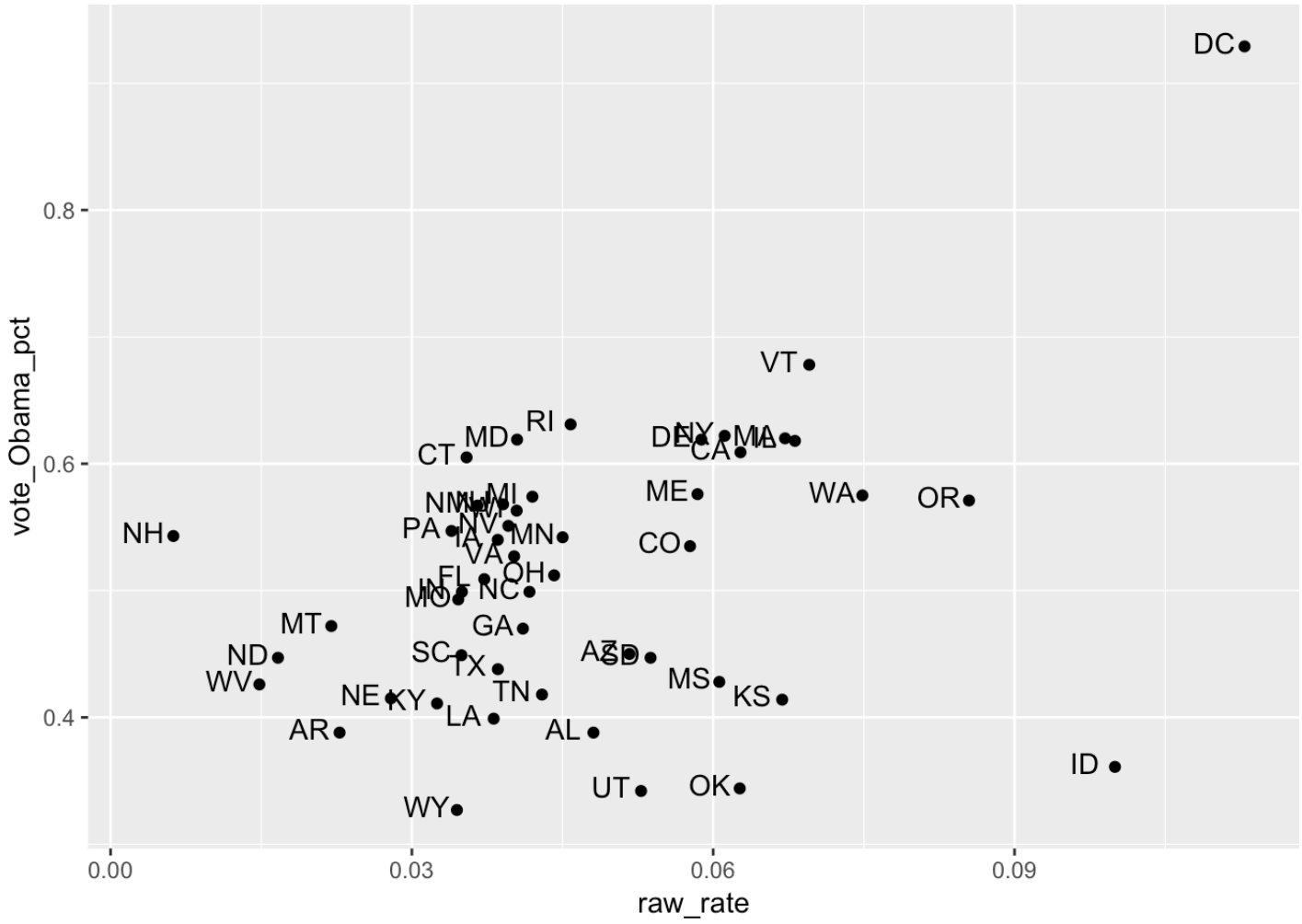
(a) graph proportion liberal in each state vs. Obama vote share

```
## data preperation
### row liberal proportion data
data <- data %>% group_by(state) %>%
            summarise(n_surv = n(),
                      n_lib = sum(ideo == "very liberal", na.rm = T),
                      raw_rate = n_lib / n_surv) %>%
            filter(! state %in% c("hawaii", "alaska"))
### obama share data
result <- result %>%  select(state,vote_Obama_pct) %>%
                     mutate(state = tolower(state)) %>%
                     filter(! state %in% c("hawaii", "alaska"))
result[,'state'][8] <- 'washington dc'
result[,'vote_Obama_pct'] <- result[,'vote_Obama_pct']/100
### combine two data together
final_data <- result %>% inner_join(data, by = "state") %>% as.data.frame()
```

```
## Warning: Column `state` joining character vector and factor, coercing into
## character vector
```

```
state_abb <- state.abb[c(-2,-11)]
state_abb_1 <- state_abb[1:7]
state_abb_2 <- state_abb[8:48]
state_abb <- c(state_abb_1,'DC',state_abb_2)
final_data[,'state'] <- state_abb

## make the plot
shift_x <- rep(0.003,49)
shift_y <- rep(0.003,49)
ggplot(final_data, aes(x= raw_rate, y=vote_Obama_pct)) + geom_point() +
    geom_text(aes(x= raw_rate-shift_x, y=vote_Obama_pct+shift_y, label=state_abb))
```



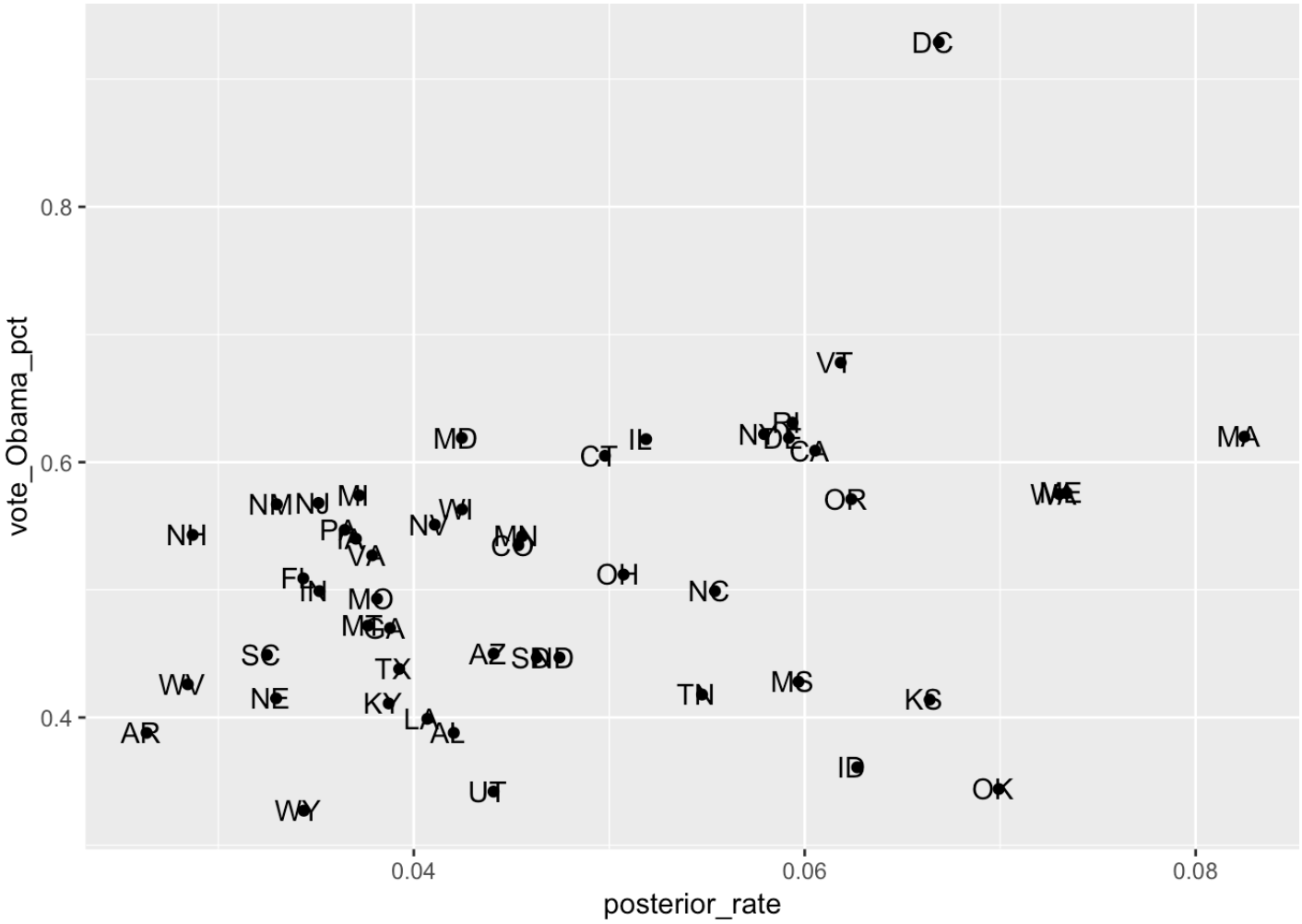
(b) graph the bayesian posterior mean in each state vs. Obama vote share

```
## likelihood: y_j ~ poisson(n_j,\theta_j): n_j means the popiu and \theta_j means th
e underlying rate of liberal
## prior: \theta_j ~ beta(\alpha,\beta)
## construct a prior distribution: based on marginal distribution of y_j follows neg_
bin(alpha, beta/n_j)
A <- mean(final_data$n_lib/final_data$n_surv)
B <- var(final_data$n_lib/final_data$n_surv)
C <- mean(1/final_data$n_surv)
beta <- A/(B-A*C)
alpha <- beta * A
cat('beta:',beta,';alpha:',alpha,'\n')
```

```
## beta: 216.9672 ;alpha: 10.25546
```

```
## thus the posterior distribution is the gamma distribution (alpha + y_j,beta + n_j)
posterior_rate <- c()
for(i in 1:dim(final_data)[1]){
    posterior_rate <- c(posterior_rate,rgamma(n=1,alpha + final_data$n_lib[i], beta +
final_data$n_surv[i]))
}
final_data$posterior_rate = posterior_rate

## make the plot
ggplot(final_data, aes(x= posterior_rate, y=vote_Obama_pct)) + geom_point()+
    geom_text(aes(x= posterior_rate-shift_x*0.1, y=vote_Obama_pct+shift_y*0.1, label=
state_abb))
```

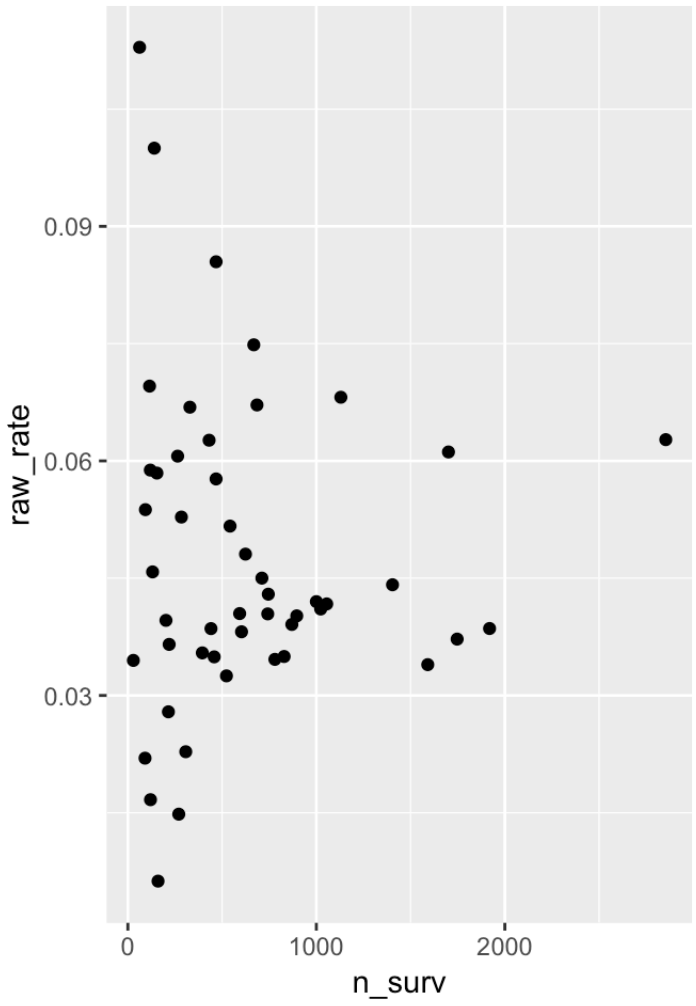


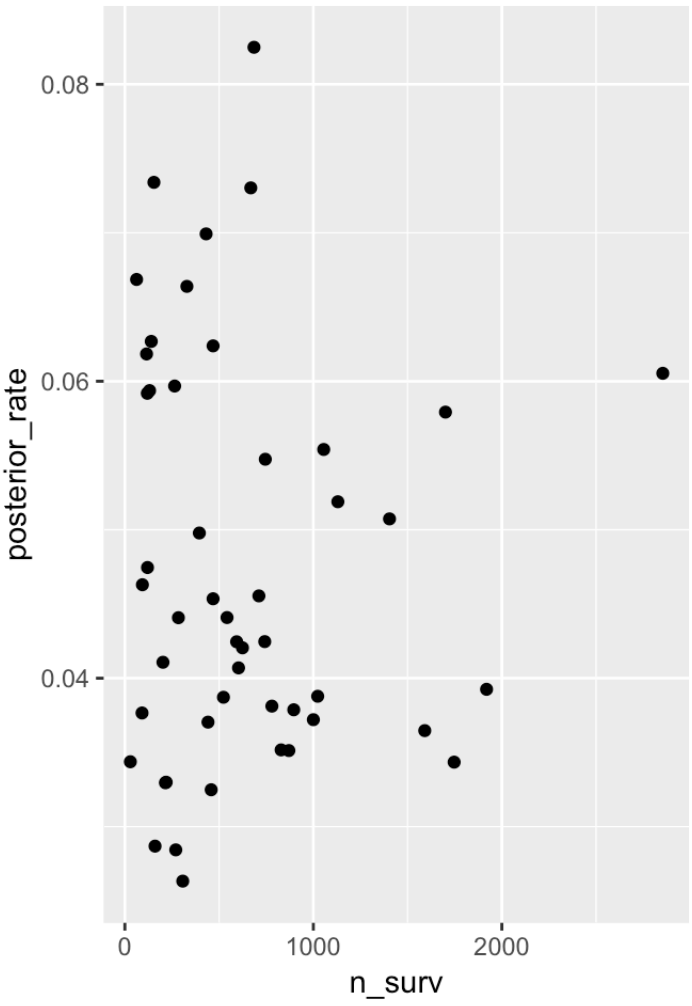
(c) repeat graph (a) and (b) using nunber of respondents in the state on x-axis.

```
plot_3 <- ggplot(final_data, aes(y= raw_rate, x=n_surv)) + geom_point()
plot_4 <- ggplot(final_data, aes(y= posterior_rate, x=n_surv)) + geom_point()
```

four plot in a single page

```
library(gridExtra)
grid.arrange(plot_3, plot_4,ncol=2)
```





2. Exercise 4.1 of BDA

(a)

$$p(y_i|\theta) \propto \frac{1}{1+(y_i-\theta)^2}$$

$$log-likelihood: log(p(\theta|\mathbf{y}) \propto log(p(\theta)) + log(p(\mathbf{y}|\theta))$$

$$= -\sum_{i=1}^{5} log(1+(y_i-\theta)^2) I_{(\theta\in[0,1])}$$

The derivative:

$$\frac{d}{d(\theta)} log(p(\theta|\mathbf{y}) = \frac{d}{d(\theta)}(-\sum_{i=1}^{5} log(1+(y_i-\theta)^2) I_{(\theta\in[0,1])})$$

$$= 2\sum_{i=1}^{5}(\frac{y_i-\theta}{1+(y_i-\theta)^2})I_{(\theta\in[0,1])}$$

The second derivative:

$$\frac{d^2}{d(\theta^2)}log(p(\theta|\mathbf{y}) = 2\frac{d^2}{d(\theta^2)}(\sum_{i=1}^{5}\frac{y_i-\theta}{1+(y_i-\theta)^2})I_{(\theta\in[0,1])}$$

$$= 2\sum_{i=1}^{5}\frac{(y_i-\theta)^2-1}{(1+(y_i-1)^2)^2}I_{(\theta\in[0,1])}$$

(b)

To get the mode, let the derivative to be zero.

$$\frac{d}{d(\theta)}log(p(\theta|\mathbf{y}) = 2\sum_{i=1}^{5}(\frac{y_i-\theta}{1+(y_i-\theta)^2})I_{(\theta\in[0,1])} = 0$$

This equation is not easy to calculate. Thus, I choose to use the Newton-Raphson Method to get the answer.

$$\hat{\theta}_1 = \hat{\theta}_0 - \frac{l'(\hat{\theta}_0)}{l''(\hat{\theta}_0)}$$

```
mlecauchy=function(x,toler=.00001){
    thetahatcurr = mean(x)
    firstderivll = 2 * sum((x-thetahatcurr)/(1+(x-thetahatcurr)^2))
    while( abs(firstderivll)){
        secondderivll= 2 * sum(((x-thetahatcurr)^2-1)/(1+(x-thetahatcurr)^2)^2)
        thetahatnew=thetahatcurr-firstderivll/secondderivll
        thetahatcurr=thetahatnew
        firstderivll=2*sum((x-thetahatcurr)/(1+(x-thetahatcurr)^2))
        }
    return(thetahatcurr)
}
y <- c(-2,-1,0,1.5,2.5)
mlecauchy(x=y)
```

```
## [1] -0.1376493
```

Thus, the posterior mode is 0.2

(c)

$$p(\theta|\mathbf{y}) \approx N(\hat{\theta}, [I(\hat{\theta})]^{-1})$$

Where

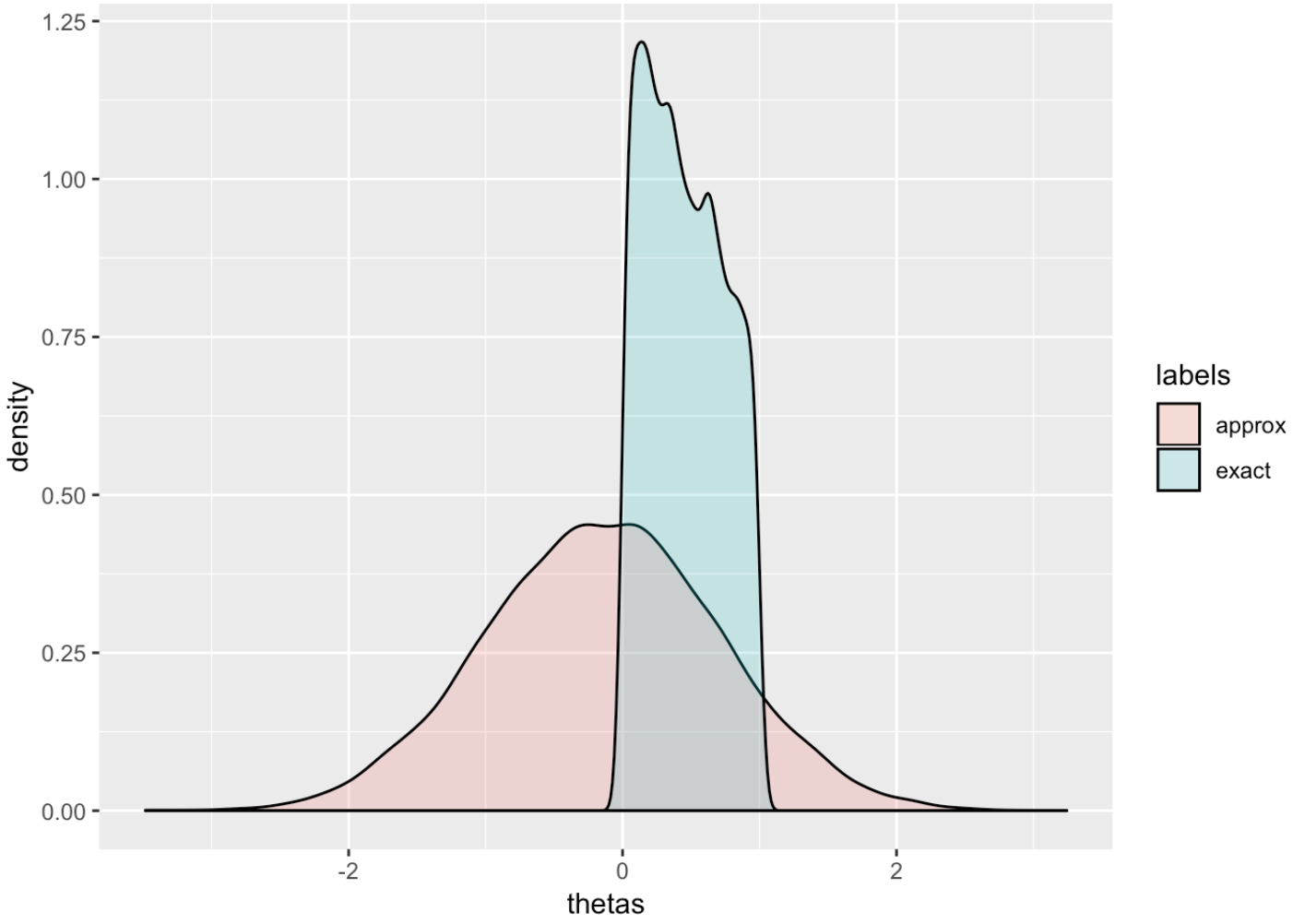
$$I(\theta) = -\frac{d^2}{d(\theta^2)} log(p(\theta|\mathbf{y}), \theta \in [0, 1]$$

```
#information:
y <- c(-2,-1,0,1.5,2.5)
thetahat <- mlecauchy(x=y)
information <- -1/ (2 * sum(((y-thetahat)^2-1)/(1+(y-thetahat)^2)^2))
information
```

```
## [1] 0.7273309
```

```
# the apporixmate distirbution is : N(0.2,1)
```

```
# draw the exact density of the posterior
dens <- function (y, theta){
    dens0 <- c()
    for (i in 1:length(theta)){
        dens0 <- c(dens0, prod (dcauchy (y, theta[i], 1)))}
    return(dens0)
}
y <- c(-2,-1,0,1.5,2.5)
step <- .01
theta <- seq(step/2, 1-step/2, step)
dens.unnorm <- dens(y,theta)
dens.norm <- dens.unnorm/(step*sum(dens.unnorm))
exact_thetas <- sample (theta, 10000, step*dens.norm, replace=TRUE)
approx_thetas <- rnorm(10000, thetahat, sd = sqrt(information))
thetas <- c(exact_thetas,approx_thetas)
labels <- c(rep('exact',length(exact_thetas)),rep('approx',length(approx_thetas)))
plot_thetas <- data.frame(thetas = thetas, labels,labels)
library(ggplot2)
ggplot(plot_thetas, aes(thetas, fill = labels)) + geom_density(alpha = 0.2)
```

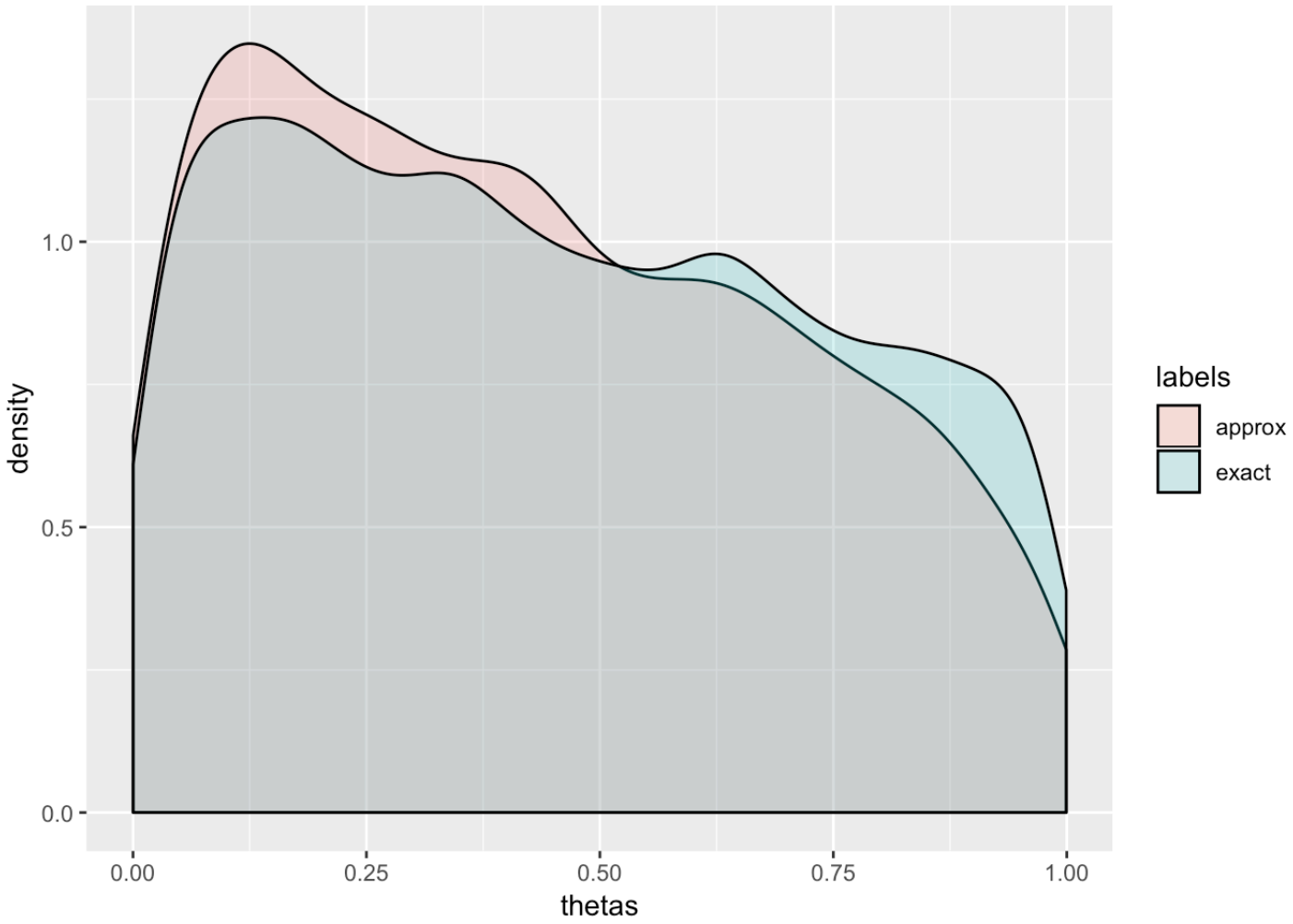


Note

The sample size in this problem is very small, thus we cannot see a good apporixmate of normal distribution to the exact distirbution. But sitll the normal indeed cover the true distribution to some extent. Since the posterior destribution should be in the parameter space that is from 0 to 1. Thus, I restandardize this.

```
approx_thetas <- approx_thetas[which((approx_thetas>0) & (approx_thetas<1))]

thetas <- c(exact_thetas,approx_thetas)
labels <- c(rep('exact',length(exact_thetas)),rep('approx',length(approx_thetas)))
plot_thetas <- data.frame(thetas = thetas, labels,labels)
library(ggplot2)
ggplot(plot_thetas, aes(thetas, fill = labels)) + geom_density(alpha = 0.2)
```



After restandardized the approximated normal distribution to the range of 0 to 1. The approximation distribution and the exact distribution is very close.