Assignment #10

Yutae Lee

2022-11-29

Problem 1

Using our knowledge of Normal-Normal conjugacy from chapter 5, we can easily derive the exact posterior.

$$p(\theta \mid \mathbf{y}) \propto exp(-\frac{1}{2}\sum_{i=1}^{n}(y_i-\theta)^2) * exp(-\frac{1}{2}\theta^2) \propto exp((-\frac{1}{2}\sqrt{n+1} * \theta - \frac{\sum_{i=1}^{n} y_i}{\sqrt{n+1}})^2)$$

From this we can get that $\mu_n = 2, \sigma_n^2 = 1/5$

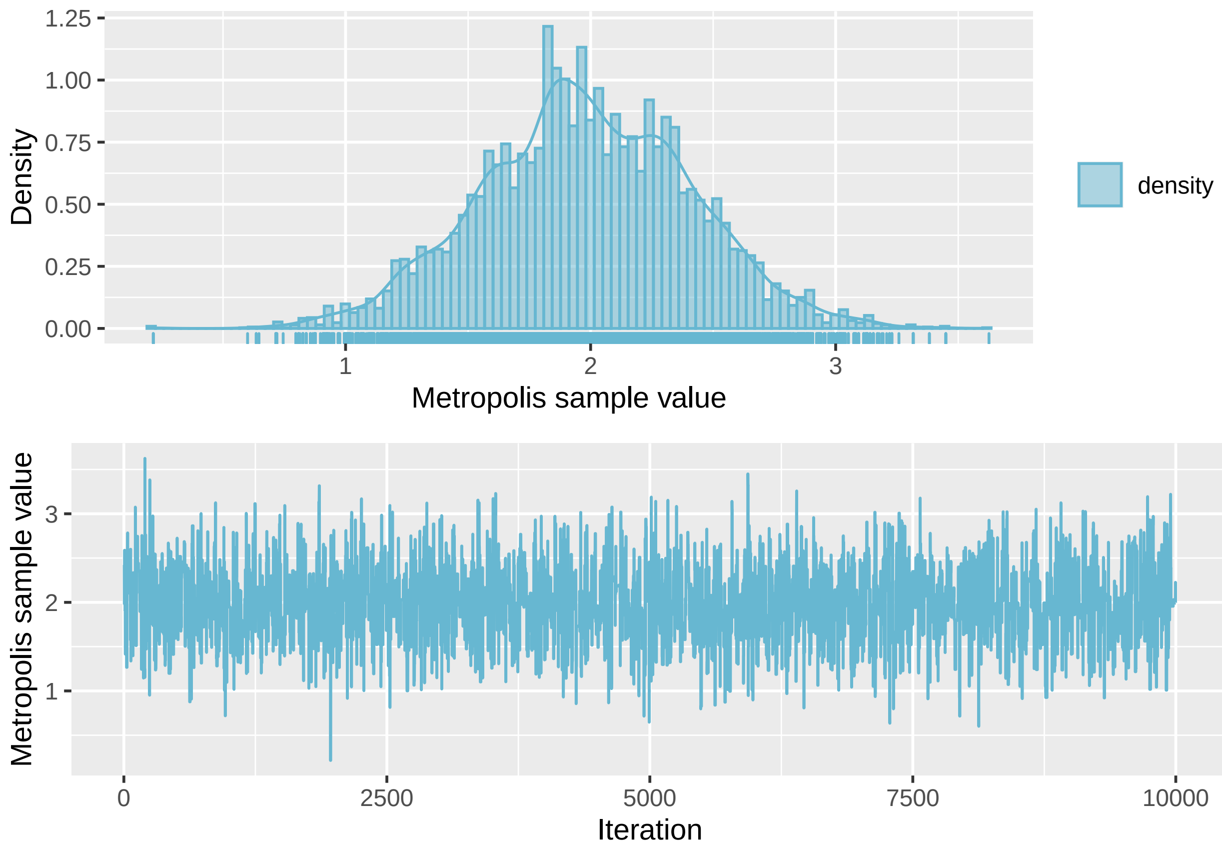
```
set.seed(32)
observation <- c(-1,0,1,10)
S = 1e4
burn_in = 500
metro_normal_samples = rep(0,S)
accepted = 0
current = 3
f <- function(mean){((-1/2)*((length(observation)-1)*var(observation) + length(observation)*(mean(obser
for (i in 1:(S+burn_in)){
  proposed = rnorm(1, mean = current, sd = 2)
  r = f(proposed) - f(current)
  if ((r >= 0 || r < 0)&& (r > log(runif(1, min = 0 , max = 1)))) {
    current = proposed
    accepted = accepted + 1
  }
  if (i >= burn_in){
    metro_normal_samples[i-burn_in] = current
  }
}

normal_df <- data.frame(metro_normal_samples)
library(ggplot2)
gg <- ggplot(normal_df,aes(x = metro_normal_samples, color = 'density')) +
  geom_histogram(aes(y = ..density..), bins = 100,  fill = '#67B7D1', alpha = 0.5) +
  geom_density(color = '#67B7D1') +
  geom_rug(color = '#67B7D1') +
  ylab("Density") +
  xlab("Metropolis sample value")  + theme(legend.title=element_blank()) +
  scale_color_manual(values = c('density' = '#67B7D1'))
gg2 <- ggplot(normal_df,
       aes(x = 1:nrow(normal_df),
           y = metro_normal_samples)) +
```

```
  geom_line(color = "#67B7D1") + ylab("Metropolis sample value") +
  xlab("Iteration")
library("cowplot")
```

```
## Warning:    'cowplot'  R   4.2.2
```

```
plot_grid(gg,gg2,
          ncol = 1, nrow = 2 )
```




```
acceptance_rate_normal = accepted / (S + burn_in)

cat('This is the accepatance rate for normal sampling:', acceptance_rate_normal)
```

```
## This is the accepatance rate for normal sampling: 0.2642857
```

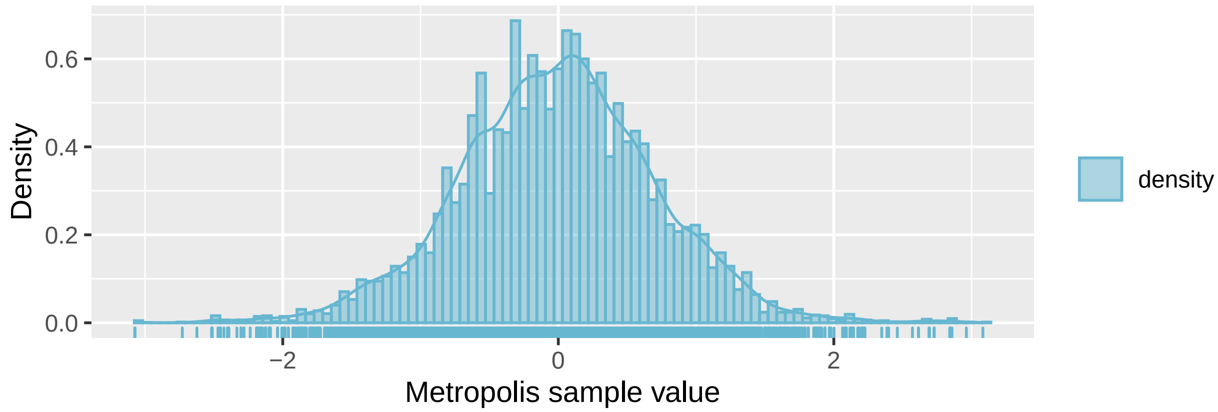
Problem 2

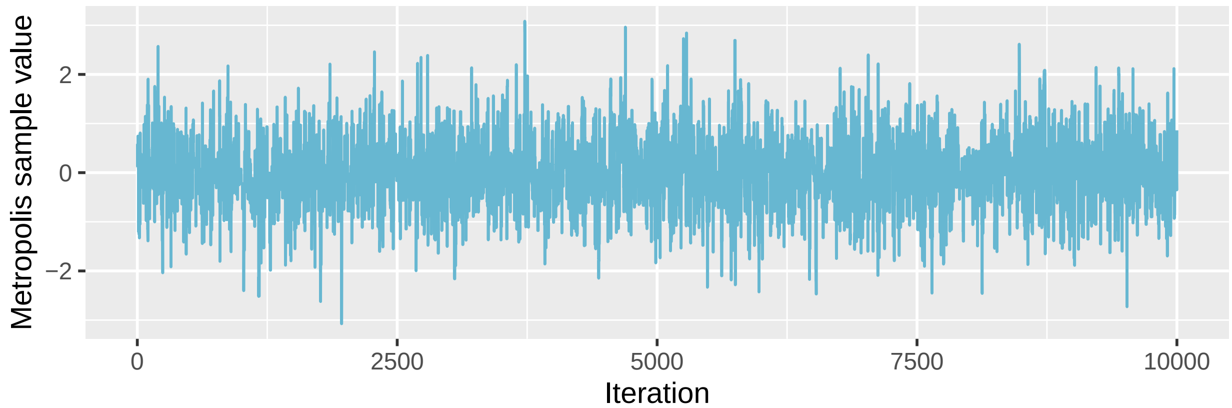
For the Cauchy:

```
set.seed(32)
observation <- c(-1,0,1,10)
S = 1e4
burn_in = 500
```

```r
metro_cauchy_samples = rep(0,S)
accepted = 0
current = 3
f <- function(mean){-log(1 + mean^2) + ((-1/2) * mean^2)}
for (i in 1:(S+burn_in)){
  proposed = rnorm(1, mean = current, sd = 2)
  r = f(proposed) - f(current)
  if ((r >= 0 || r < 0)&& (r > log(runif(1, min = 0 , max = 1)))) {
    current = proposed
    accepted = accepted + 1
  }
  if (i >= burn_in){
    metro_cauchy_samples[i-burn_in] = current
  }
}

cauchy_df <- data.frame(metro_cauchy_samples)
library(ggplot2)
gg3 <- ggplot(cauchy_df,aes(x = metro_cauchy_samples, color = 'density')) +
  geom_histogram(aes(y = ..density..), bins = 100,  fill = '#67B7D1', alpha = 0.5) +
  geom_density(color = '#67B7D1') +
  geom_rug(color = '#67B7D1') +
  ylab("Density") +
  xlab("Metropolis sample value")  + theme(legend.title=element_blank()) +
  scale_color_manual(values = c('density' = '#67B7D1'))
gg4 <- ggplot(cauchy_df,
              aes(x = 1:nrow(cauchy_df),
                  y = metro_cauchy_samples)) +
  geom_line(color = "#67B7D1") + ylab("Metropolis sample value") +
  xlab("Iteration")
library("cowplot")
plot_grid(gg3,gg4,
          ncol = 1, nrow = 2)
```






```
acceptance_rate_cauchy = accepted / (S + burn_in)

cat('This is the accepatance rate for cauchy sampling:', acceptance_rate_cauchy)
```

```
## This is the accepatance rate for cauchy sampling: 0.3805714
```

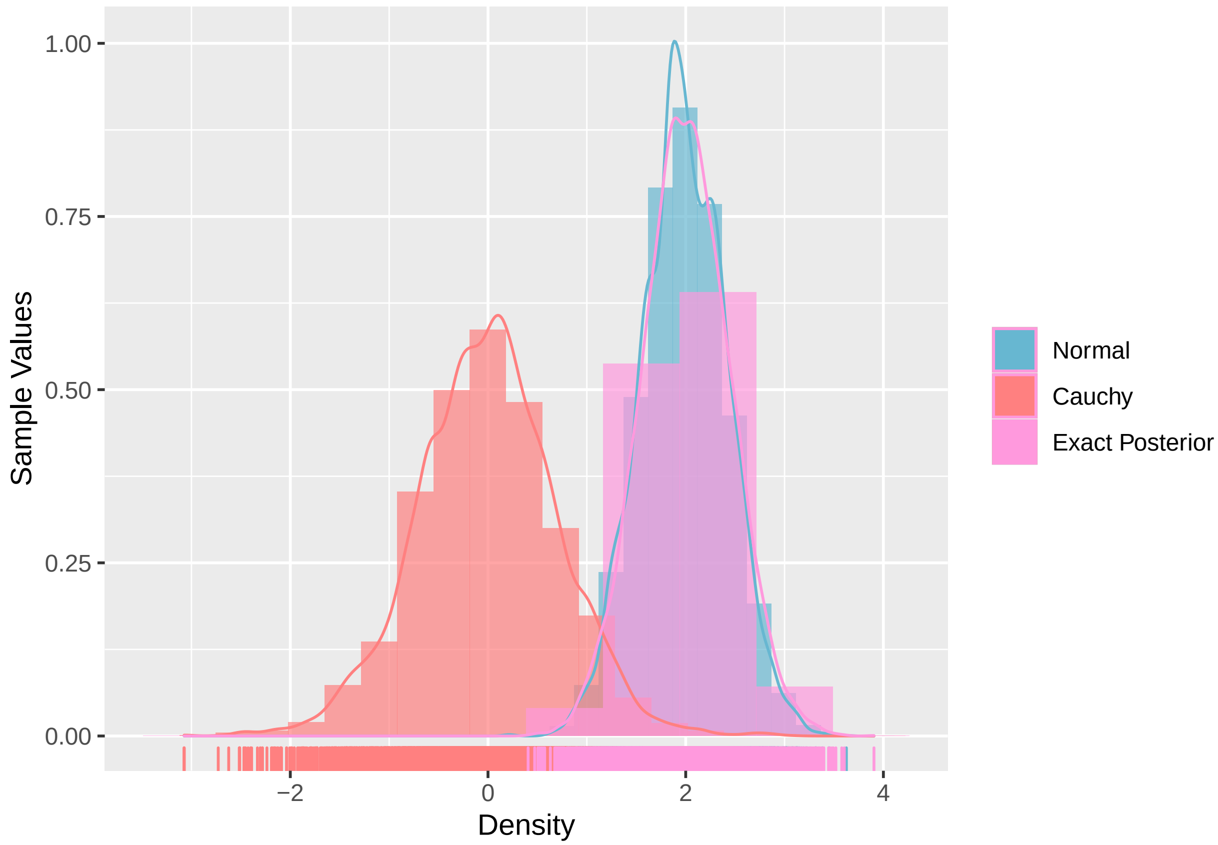
Problem 3

```
exact_posterior = rnorm(10000, mean = 2, sd = 1/sqrt(5))
group_labels = c('Normal', 'Cauchy', 'Exact Posterior')

df = data.frame(metro_normal_samples, metro_cauchy_samples, exact_posterior)

gg <- ggplot(df,aes() ) +
  geom_histogram(aes(x = metro_normal_samples, y = ..density.., fill = '#67B7D1'), alpha = 0.7, bins =
  geom_histogram(aes(x = metro_cauchy_samples, y = ..density.., fill = '#ff8080'), alpha = 0.7, bins =
  geom_histogram(aes(x = exact_posterior, y = ..density.., fill = '#ff99dd'), alpha = 0.7, bins = 10) +
  geom_density(aes(x = metro_normal_samples),color = '#67B7D1') +
  geom_density(aes(x = metro_cauchy_samples),color = '#ff8080') +
  geom_density(aes(x = exact_posterior),color = '#ff99dd') +
  geom_rug(aes(x = metro_normal_samples),color = '#67B7D1') +
  geom_rug(aes(x = metro_cauchy_samples),color = '#ff8080') +
  geom_rug(aes(x = exact_posterior),color = '#ff99dd') +
```

```r
  theme(legend.title=element_blank()) +
  scale_fill_identity(labels = c('Normal', 'Cauchy', 'Exact Posterior'),
                      guide = "legend")  +
  labs(x = 'Density',
       y = 'Sample Values')
gg
```



We can see that because of the outlier the posterior has been pulled from its true mean which supposed to be 0. But, we can still see that metropolis algorithm has apropriately aligned the sample for its true posterior.