STA 602. HW05

Yicheng Shen

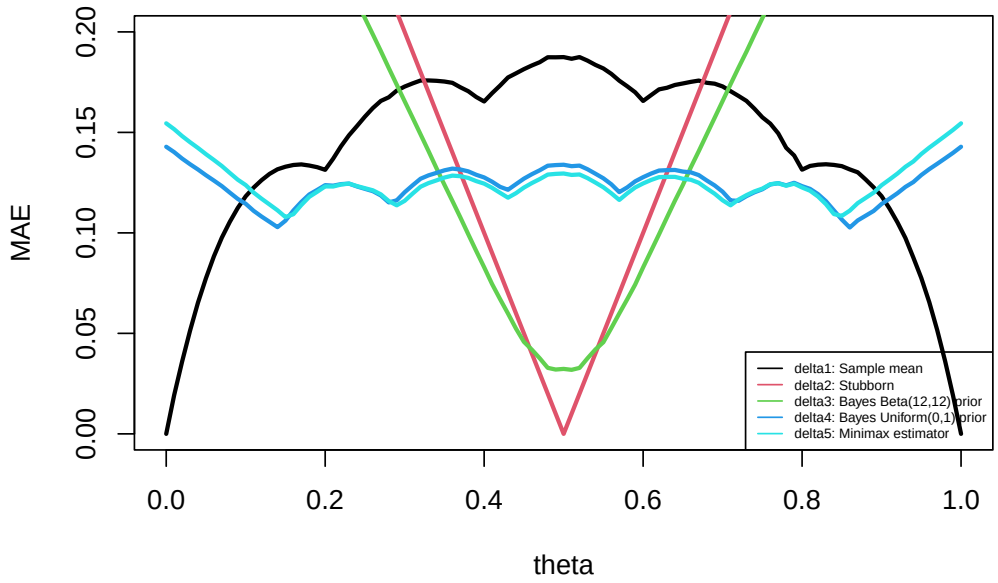
9/30/2022

3 from Last time.

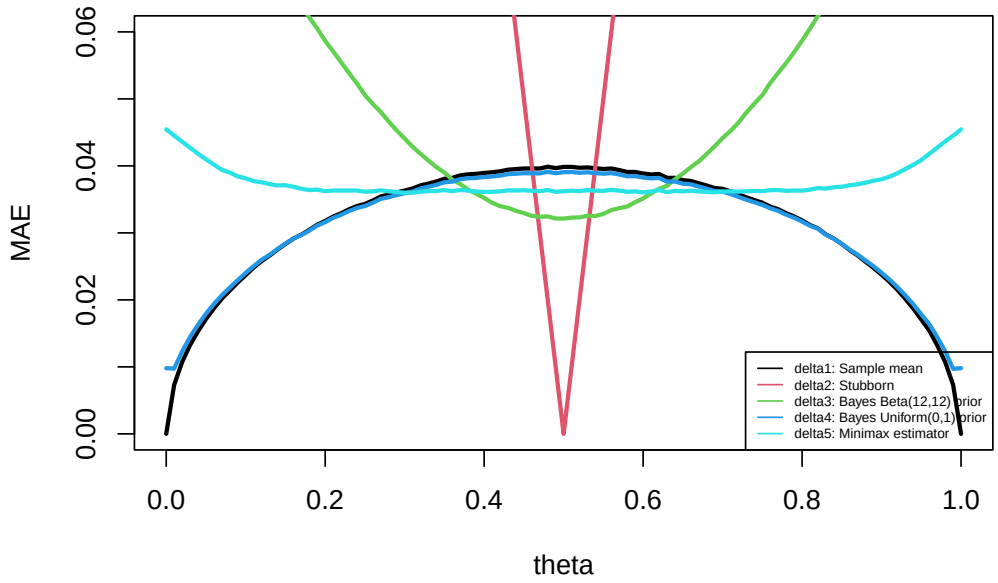
The Monte Carlo is done below for $n = 5$ and $n = 100$. Although the shapes of MAE are roughly the same as MSE, it can be noticed that MAEs for $\delta_1, \delta_4, \delta_5$ have several local minimums.

The minimax estimator behaves similarly, although when θ is near the edge values, there seems to be a rise for MAE of δ_5. Nevertheless, δ_5 is still able to maintain relatively low MAE throughout θ values. It is, however, not necessarily the minimax estimator now due to the rise of MAE.

```
n <- 5
S <- 100000
theta <- seq(0, 1, by = 0.01)

sample <- vector(mode = "list", length(theta))
MAE = MAE_2 = MAE_3 = MAE_4 = MAE_5 = vector(mode = "list", length(theta))

for (i in seq_along(theta))
{
  sample[[i]] <- rbinom(S, size = n, theta[[i]])
  MAE[[i]] <- mean(abs(sample[[i]]/n - theta[[i]]))
  MAE_2[[i]] <- mean(abs(1/2 - theta[[i]]))
  MAE_3[[i]] <- mean(abs( (sample[[i]]+12)/(n+24) - theta[[i]]))
  MAE_4[[i]] <- mean(abs( (sample[[i]]+1)/(n+2) - theta[[i]]))
  MAE_5[[i]] <- mean(abs( (sample[[i]]+sqrt(n)/2)/(n+sqrt(n)) - theta[[i]]))
}
plot(theta, MAE, type = "l", col= 1, lwd = 2.5, ylim = c(0, 0.2))
lines(theta, MAE_2, type = "l", col= 2, lwd = 2.5)
lines(theta, MAE_3, type = "l", col= 3, lwd = 2.5)
lines(theta, MAE_4, type = "l", col= 4, lwd = 2.5)
lines(theta, MAE_5, type = "l", col= 5, lwd = 2.5)
legend(x = "bottomright", legend=c("delta1: Sample mean",
                           "delta2: Stubborn",
                           "delta3: Bayes Beta(12,12) prior",
                           "delta4: Bayes Uniform(0,1) prior",
                           "delta5: Minimax estimator"),
       col = 1:5, lty = 1, cex = 0.5)
```




```r
n <- 100
S <- 100000
theta <- seq(0, 1, by = 0.01)

sample <- vector(mode = "list", length(theta))
MAE = MAE_2 = MAE_3 = MAE_4 = MAE_5 = vector(mode = "list", length(theta))

for (i in seq_along(theta))
{
  sample[[i]] <- rbinom(S, size = n, theta[[i]])
  MAE[[i]] <- mean(abs(sample[[i]]/n - theta[[i]]))
  MAE_2[[i]] <- mean(abs(1/2 - theta[[i]]))
  MAE_3[[i]] <- mean(abs( (sample[[i]]+12)/(n+24) - theta[[i]]))
  MAE_4[[i]] <- mean(abs( (sample[[i]]+1)/(n+2) - theta[[i]]))
  MAE_5[[i]] <- mean(abs( (sample[[i]]+sqrt(n)/2)/(n+sqrt(n)) - theta[[i]]))
}
plot(theta, MAE, type = "l", col= 1, lwd = 2.5, ylim = c(0, 0.06))
lines(theta, MAE_2, type = "l", col = 2, lwd = 2.5)
lines(theta, MAE_3, type = "l", col = 3, lwd = 2.5)
lines(theta, MAE_4, type = "l", col = 4, lwd = 2.5)
lines(theta, MAE_5, type = "l", col = 5, lwd = 2.5)
legend(x = "bottomright", legend=c("delta1: Sample mean",
                           "delta2: Stubborn",
                           "delta3: Bayes Beta(12,12) prior",
                           "delta4: Bayes Uniform(0,1) prior",
                           "delta5: Minimax estimator"),
       col = 1:5, lty = 1, cex = 0.5)
```



4.1

First of all, assuming a uniform prior means a Beta(1, 1) prior here. With the sampling data having $(X = 30, n = 50)$, we can derive the posterior distribution to be Beta$(31, 21)$.

From Ex 3.1 (e), we have also shown that the posterior of that is Beta$(58, 44)$. Using Monte Carlo below, we can approximate that is $Pr(\theta_1 < \theta_2 | \text{the data and prior}) \approx 0.6354$

```
set.seed(8848)  # only have some variation
theta1.mc = rbeta(5000, 58, 44)
theta2.mc = rbeta(5000, 31, 21)
mean(theta1.mc < theta2.mc)
```

```
## [1] 0.6354
```

4.2

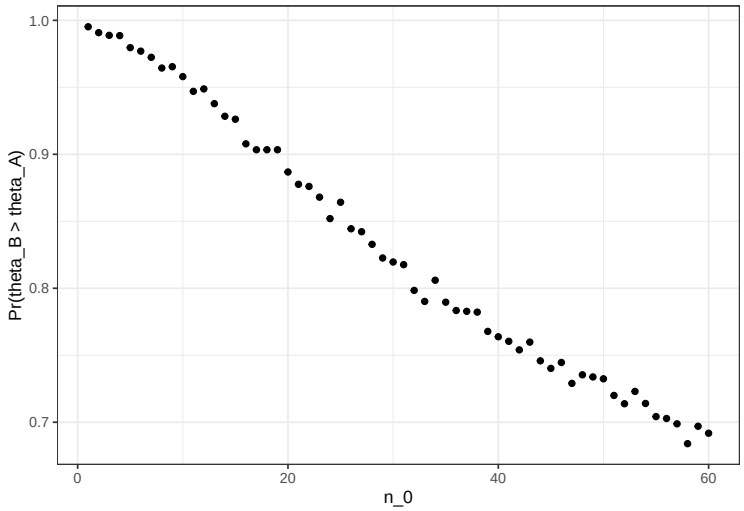
(a) We have shown in 3.3 that for θ_A, its posterior distribution is Gamma(120+117,10+10) = Gamma(237,20) and for θ_B, its posterior distribution is Gamma(12+113,1+13) = Gamma(125,14). So by Monte Carlo below, $Pr(\theta_B < \theta_A | y_A, y_B) \approx 0.9964$

```
set.seed(1999)
theta.a = rgamma(5000, 237, 20)
theta.b = rgamma(5000, 125, 14)
mean(theta.b < theta.a)
```

```
## [1] 0.9964
```

(b) The factor n_0 does have an effect on the event $\{\theta_B > \theta_A\}$. However, even when n_0 goes from small to quite large values, the probability of $\theta_B > \theta_A$ is still high, so its effect is not substantial. From the plot below, the probability and n_0 seem to have a linear relationship with negative slope.

```
n0 <- 1:60
pr_b_a <- list()
set.seed(1949)
for (i in seq_along(n0))
{ pr_b_a[i] <- mean(rgamma(5000, (12 * n0[i]) + 113, n0[i] + 13) < rgamma(5000, 237, 20)) }
ggplot() + geom_point(aes(x=n0, y=unlist(pr_b_a))) + labs(x="n_0", y="Pr(theta_B > theta_A)")
```

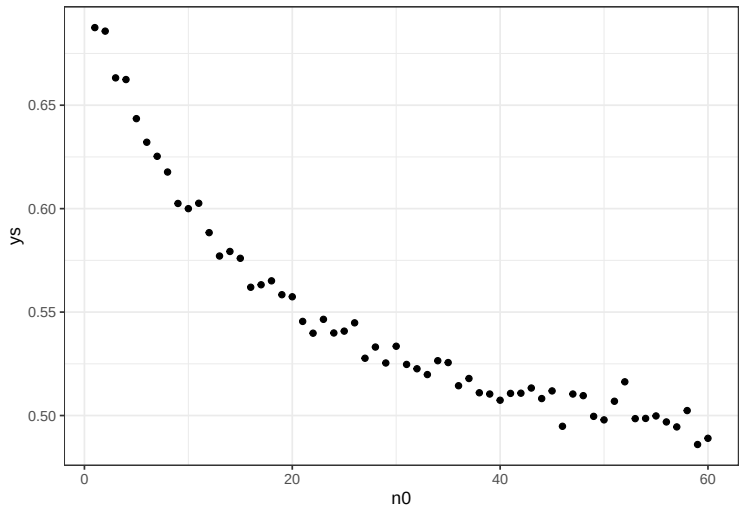


(c) The factor n_0 has a different and much stronger effect on the event $\{\tilde{Y}_b > \tilde{Y}_A\}$. Initially, the probability can be as high as 0.7. As n_0 goes to large values (approximately over 50), the probability of $\{\tilde{Y}_b > \tilde{Y}_A\}$ begins to fall below 0.5. From the plot below, the probability and n_0 seem to have a nonlinear relationship.

```
set.seed(1989)
S <- 10000
theta.a = rgamma(S, 237, 20)
theta.b = rgamma(S, 125, 14)
mean(rpois(S, theta.b) <   rpois(S, theta.a))
```

```
## [1] 0.7001
```

```
ys = sapply(n0, function(n) {
  theta.a = rgamma(S, 237, 20)
  theta.b = rgamma(S, (12 * n) + 113, n + 13)
  mean(rpois(S, theta.b) <   rpois(S, theta.a))
})
qplot(n0, ys, geom = c('point'))
```

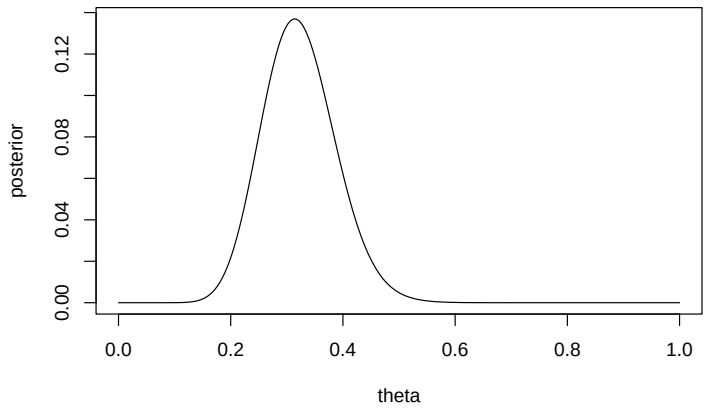


4.4

(a) The posterior density is proportional to

$$p(y|\theta)p(\theta) \propto \left[\binom{43}{15}\theta^{15}(1-\theta)^{28}\right] \times \frac{1}{4}\frac{\Gamma(10)}{\Gamma(2)\Gamma(8)}[3\theta(1-\theta)^7 + \theta^7(1-\theta)]$$

```
theta <- seq(from = 0, to = 1, length.out = 10000)
theta_pdf <- 0.25 * gamma(10) / (gamma(2) * gamma(8)) * (3 * theta * (1 - theta) ^ 7 + theta ^ 7 * (1 -
likelihood <- choose(43, 15) * theta ^ 15 * (1 - theta) ^ 28
posterior <- theta_pdf * likelihood
plot(theta, posterior, type = "l")
```




```
theta[which(cumsum(posterior) / sum(posterior) >= 0.025)[1]]
```

```
## [1] 0.2036204
```

```
theta[which(cumsum(posterior) / sum(posterior) >= 0.975)[1]-1]
```

```
## [1] 0.4576458
```

For this posterior density, using discrete approximation gives a rough 95% CI from 0.2036204 to 0.4576458.

(b)

```
w <- 0.75*beta(2+15, 8+28)/beta(2,8) /
  (0.75*beta(2+15, 8+28)/beta(2,8)  + 0.25*beta(8+15, 2+28)/beta(8,2))
z <- list()
for (i in 1:10000) {
x <- rbinom(1,1,w)
if (x == 1) {z[[i]] = rbeta(1,17,36)}
else {z[[i]] = rbeta(1,23,30)}
}
quantile(unlist(z), c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.2059760 0.4607997
```

```
density(unlist(z))
```

```
##
## Call:
##  density.default(x = unlist(z))
##
## Data: unlist(z) (10000 obs.);    Bandwidth 'bw' = 0.009314
##
##        x                y
##  Min.   :0.0966   Min.   :0.000049
##  1st Qu.:0.2361   1st Qu.:0.034285
##  Median :0.3755   Median :0.632220
##  Mean   :0.3755   Mean   :1.790876
##  3rd Qu.:0.5150   3rd Qu.:3.552290
##  Max.   :0.6544   Max.   :5.985839
```

The CI from Monte Carlo is (0.2048947, 0.4587787). Compared with the previous one, the two quantile-based CIs are in fact very similar.

4.5

(a) It can be proved that

$$\begin{aligned} p(\theta|y_i, x_i) &\propto \frac{b^a}{\Gamma(a)}\theta^{a-1}e^{-b\theta} \times \prod_{i=1}^{n} \frac{(\theta X_i)^{Y_i} e^{-(\theta X_i)}}{Y_i!} \\ &\propto \theta^{a-1}e^{-b\theta}\theta^{\sum Y_i}e^{-\theta\sum X_i} \\ &\propto \theta^{a+\sum Y_i-1}e^{-(b+\sum X_i)\theta} \\ &\sim \text{Gamma}(a+\sum Y_i, b+\sum X_i) \end{aligned}$$

(b)

```
cancer_react <- read.table("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/cancer_react.dat", header=TR
cancer_noreact <- read.table("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/cancer_noreact.dat",
                             header=TRUE)
colSums(cancer_noreact)
```

```
##    x    y
## 1037 2285
```

```
colSums(cancer_react)
```

```
##   x   y
##  95 256
```

Therefore, we know that $\theta_1 \sim \text{Gamma}(a + 2285, b + 1037)$ and $\theta_1 \sim \text{Gamma}(a + 256, b + 95)$.

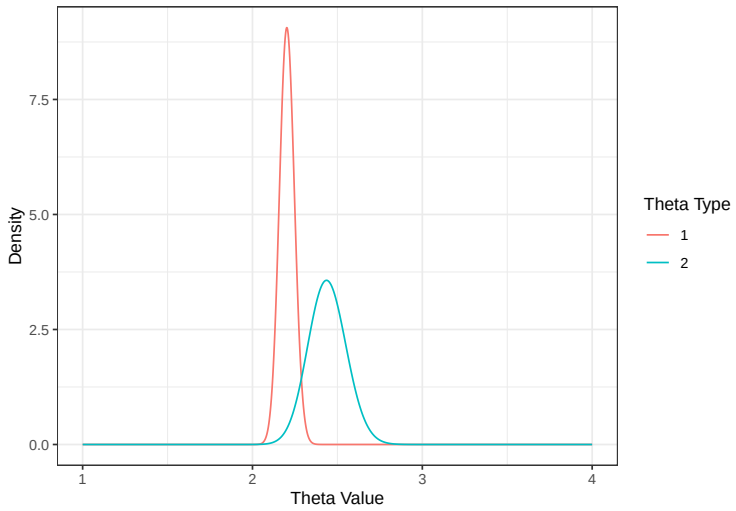
(c) I built a function to streamline the following problems:

```
compute_opinion <- function(a1, b1, a2, b2, S) {
  a1_post = a1 + sum(cancer_noreact$y)
  b1_post = b1 + sum(cancer_noreact$x)
  a2_post = a2 + sum(cancer_react$y)
  b2_post = b2 + sum(cancer_react$x)

  theta1_mean = a1_post / b1_post; theta2_mean = a2_post / b2_post
  theta1_lb = qgamma(0.025, a1_post, b1_post); theta1_up = qgamma(0.975, a1_post, b1_post)
  theta2_lb = qgamma(0.025, a2_post, b2_post); theta2_up = qgamma(0.975, a2_post, b2_post)
  pr_theta_1_2 = mean(rgamma(S, a2_post, b2_post) > rgamma(S, a1_post, b1_post))
  report_data = cbind(c(theta1_mean, theta2_mean), c(theta1_lb, theta2_lb), c(theta1_up, theta2_up))

  colnames(report_data) = c('Expectation / Posterior Mean', '95% CI lower bound', '95% CI upper bound')
  rownames(report_data) = c('theta1', 'theta2')
  x = seq(from = 1, to = 4, length.out = 1000)
  post_graph = data.frame(x = c(x, x),
                          y = c(dgamma(x, a1_post, b1_post), dgamma(x, a2_post, b2_post)),
                          theta = rep(c(1, 2), each = 1000)) %>%
                          ggplot() + geom_line(aes(x, y, col = factor(theta))) +
                          labs(x = "Theta Value", y = "Density", color = "Theta Type")
  print(report_data)
  cat("Pr(theta2 > theta1 | data):", pr_theta_1_2)
  print(post_graph)
}
```

Opinion 1: posterior mean, 95% CI and probability are shown in the output below

```
compute_opinion(a1 = 220, b1 = 100, a2 = 220, b2 = 100, S = 5000)
```

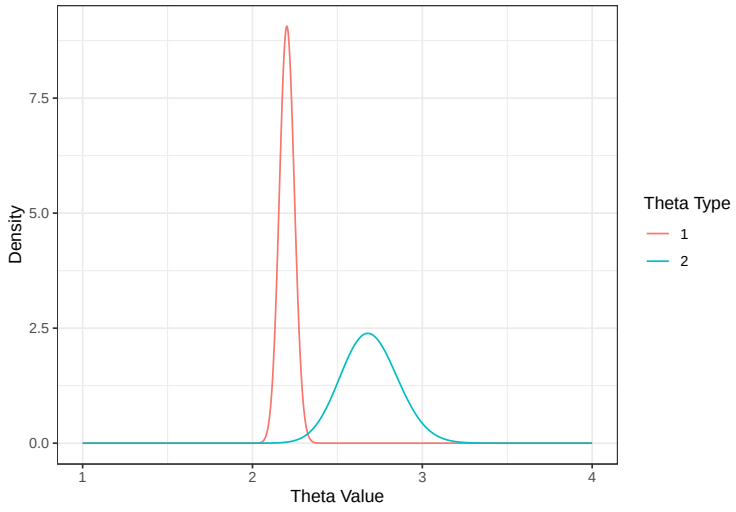
```
##        Expectation / Posterior Mean 95% CI lower bound 95% CI upper bound
## theta1                     2.203166          2.117726          2.290273
## theta2                     2.441026          2.226633          2.665131
## Pr(theta2 > theta1 | data): 0.9788
```



Opinion 2:

```
compute_opinion(a1 = 220, b1 = 100, a2 = 2.2, b2 = 1, S = 5000)
```

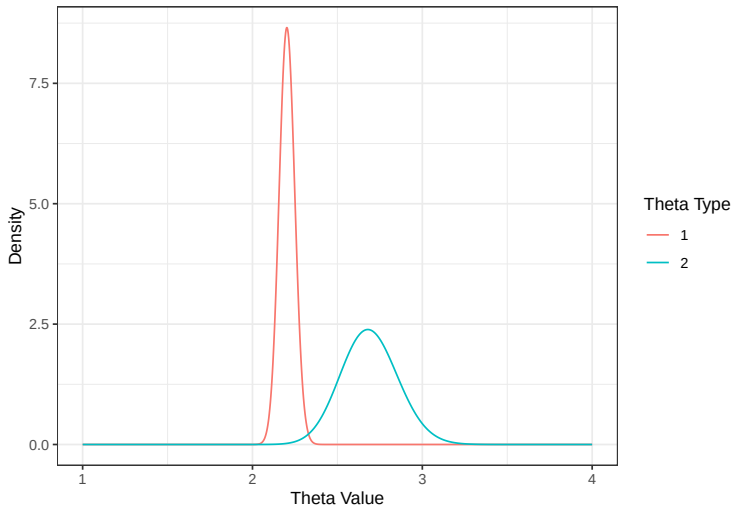
```
##        Expectation / Posterior Mean 95% CI lower bound 95% CI upper bound
## theta1                     2.203166           2.117726           2.290273
## theta2                     2.689583           2.371497           3.027397
## Pr(theta2 > theta1 | data): 0.9984
```



Opinion 3:

```
compute_opinion(a1 = 2.2, b1 = 1, a2 = 2.2, b2 = 1, S = 5000)
```

```
##        Expectation / Posterior Mean 95% CI lower bound 95% CI upper bound
## theta1                     2.203468           2.114081           2.294680
## theta2                     2.689583           2.371497           3.027397
## Pr(theta2 > theta1 | data): 0.9974
```



Comments: the posterior outcomes from opinion 1 is different from the rest two opinions. This should be mainly because we have limited amount of sampled data, especially for reactor type counties. Therefore the prior belief plays an important role. While opinion 1 assumes that cancer rates for both types of counties are similar to the average rates across all counties from previous years, opinion 2 and 3 assume that there could be differences. Because of this difference in beliefs, the difference between posterior means of θ_1, θ_2 is smaller from opinion 1.

Opinions 2 and 3 have very similar outcomes. The difference between them lies in their prior beliefs of cancer rates in this year for nonreactor counties. However, we do have many more sampled data from nonreactor counties and these data do suggest that the θ value may be close to 2.2. As a result, the posteriors from opinions 2 and 3 behave similarly.

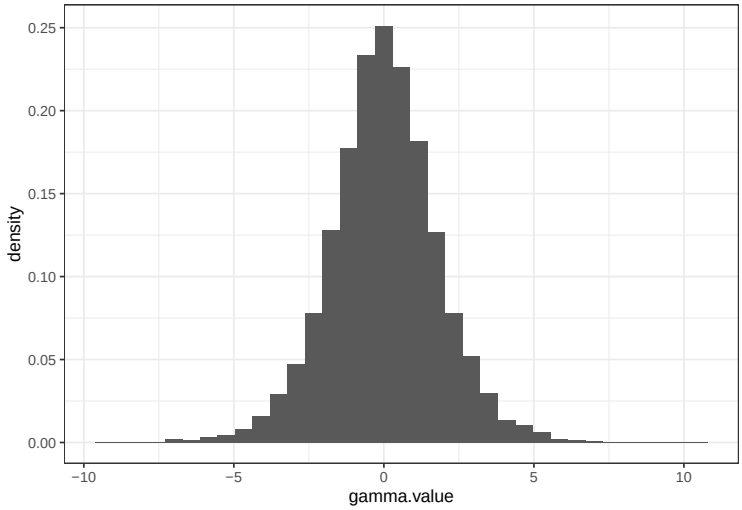
(d) The population size is taken into account as a variable as X_i in the sampling model (which is why we can say θ is a fatality rate, not counts). Since cancer is not a contagious disease, we do not have any reason to include the effect of population size when analyzing cancer fatality rate.

(e) It is possible that climates, geography, food sources and other factors can all affect cancer fatality rate. Some neighboring counties that have similar climate or food supplies could end up with similar fatality, hence non-independent θ_1, θ_2.

Regarding how, it is possible that we can use a joint distribution $p(\theta_1, \theta_2)$ to describe our prior beliefs while maintaining a dependent relation between the two.

4.6

```
set.seed(2345)
theta.mc <- runif(10000)
gamma.mc <- sapply(theta.mc, function(theta) log(theta / (1 - theta)))
ggplot(data.frame(gamma.value = gamma.mc), aes(x = gamma.value, y = ..density..)) +
  geom_histogram(bins = 35)
```



The prior distribution for γ is centered around zero and looks like to be normally distributed. So this does represent some prior information about the value of γ instead of "no information at all".