Stat 343: Intro to Monte Carlo methods for Bayesian inference

Cosmological Microwave Background (CMB)

This example is taken from Marin and Robert (2007). Here's a quote from them describing the figure below, also from them:

> 'Figure 2.2 is an image (in the spectral domain) of the "cosmological microwave background" (CMB) in a region of the sky: More specifically, this picture represents the electromagnetic radiation from photons dating back to the early ages of the universe, a radiation often called "fossil light," that dates back to a few hundred thousand years after the Big Bang (Chown, 1996). The grey levels are given by the differences in apparent temperature from the mean temperature and as stored in `cmb`.
>
> For astrophysical (or rather cosmological) reasons too involved to be detailed here, the repartition of the sectrum is quite isotropic (that is, independent of direction) and normal. In fact, if we treat each temperature difference in Figure 2.2 as an independent realization, the histogram of these differences ... provides a rather accurate representation of the distribution of these temperatures...'

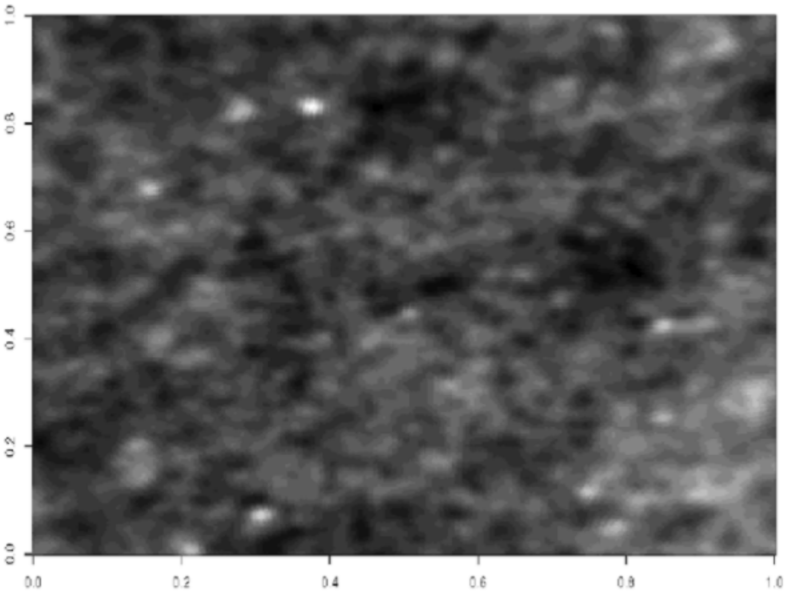
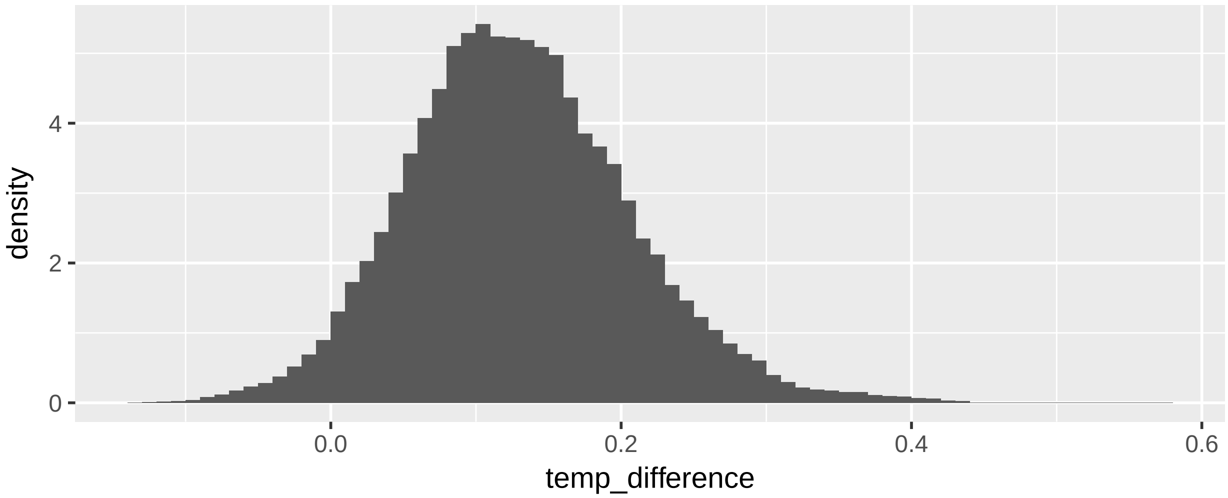


Fig. 2.2. Dataset CMBdata: Spectral image of the cosmological microwave background (CMB) of the universe. (The darker the pixel, the higher the temperature difference from the mean temperature.)

The code below reads in the data and makes an initial plot:

```r
library(tidyverse)

cmb <- read_csv("http://www.evanlray.com/stat343_s2018/data/bayesian_core/CMBdata.txt",
    col_names = FALSE)
names(cmb) <- "temp_difference"

ggplot(data = cmb, mapping = aes(x = temp_difference)) +
  geom_histogram(center = 0.005, binwidth = 0.01, mapping = aes(y = ..density..))
```



Model

It appears that a normal model would be reasonable for these data. To be formal, let $X_1, \dots, X_n$ denote the $n = 640000$ temperature differences. We model these as independent, with each

$X_i \sim \text{Normal}(\mu, \sigma^2)$

This model has two parameters: μ and σ^2. We will use the following prior distributions for these parameters:

An improper, non-informative prior for μ:

$f(\mu) = 1$

A Gamma$(2, 1)$ prior for σ^2:

$f(\sigma^2|\alpha = 2, \beta = 1) = \frac{\beta^\alpha}{\Gamma(\alpha)}(\sigma^2)^{\alpha-1}e^{-\beta x}$

The probability density function of the posterior distribution for μ and σ^2 is therefore equal to a constant c times the prior pdfs times the data model pdf:

$$\begin{aligned}f(\mu, \sigma^2|\alpha = 2, \beta = 1, x_1, \dots, x_n) &= c \cdot f(\mu) \cdot f(\sigma^2|\alpha = 2, \beta = 1) \cdot f(x_1, \dots, x_n|\mu, \sigma^2) \\ &= c \cdot 1 \cdot \frac{\beta^\alpha}{\Gamma(\alpha)}(\sigma^2)^{\alpha-1}e^{-\beta x} \cdot \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{-1}{2\sigma^2}(x_i - \mu)^2\right]\end{aligned}$$

Big idea:

The posterior distribution is involved. Rather than trying to understand it analytically, let's take a sample from the joint posterior distribution of μ and σ^2, and use that to represent the posterior distribution.

Samples from the posterior distribution

No need to understand this code for now. The point is that it draws a sample of size 10,000 from the posterior distribution.

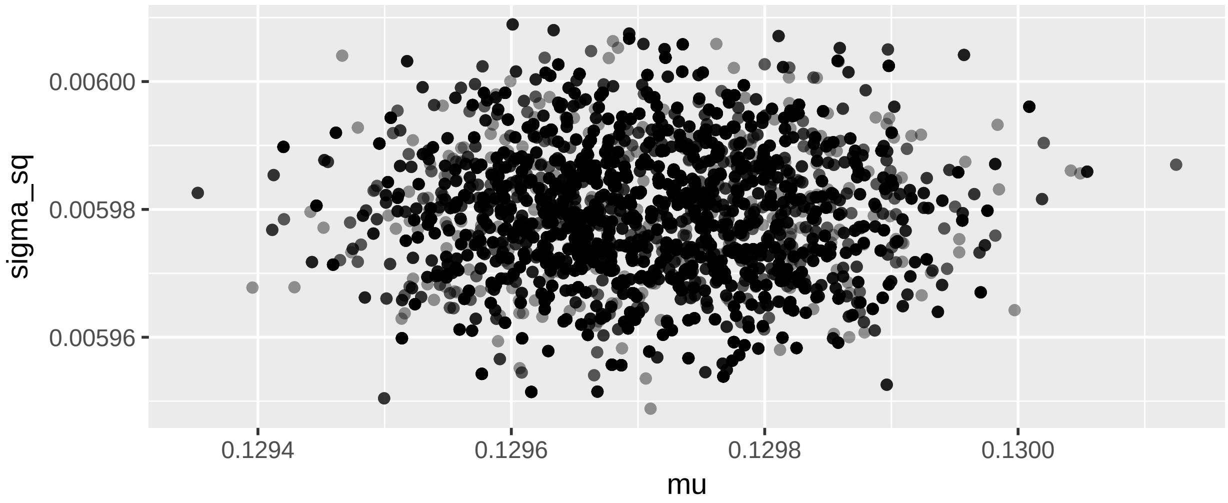
```
sample_size <- 10000
burn_in <- 100
theta_posterior_sample <- data.frame(
  mu = rep(NA, sample_size + burn_in),
```

```r
  log_sigma_sq = rep(NA, sample_size + burn_in),
  sigma_sq = rep(NA, sample_size + burn_in)
)
theta_posterior_sample$mu[1] <- mean(cmb$temp_difference)
theta_posterior_sample$log_sigma_sq[1] <- log(var(cmb$temp_difference))
theta_posterior_sample$sigma_sq[1] <- var(cmb$temp_difference)

for(i in seq(from = 2, to = sample_size + burn_in)) {
    cat(i)
    cat("\n")
    # Generate a proposal for the next value of theta, from a
    # Uniform(previous_theta - 0.1, previous_theta + 0.1) distribution
    previous_mu <- theta_posterior_sample$mu[i-1]
    previous_log_sigma_sq <- theta_posterior_sample$log_sigma_sq[i-1]
    mu_proposal <- runif(1, previous_mu - .001, previous_mu + .001)
    log_sigma_sq_proposal <- runif(1, previous_log_sigma_sq - .001, previous_log_sigma_sq + .001)

    # calculate probability of accepting the proposal
    log_r_num <- dgamma(exp(log_sigma_sq_proposal), shape = 2, rate = 1, log = TRUE) +
        sum(dnorm(cmb$temp_difference,
                  mean = mu_proposal,
                  sd = sqrt(exp(log_sigma_sq_proposal)),
                  log = TRUE))
    log_r_denom <- dgamma(exp(previous_log_sigma_sq), shape = 2, rate = 1, log = TRUE) +
        sum(dnorm(cmb$temp_difference,
                  mean = previous_mu,
                  sd = sqrt(exp(previous_log_sigma_sq)),
                  log = TRUE))

    r <- min(1,
        exp(log_r_num - log_r_denom))

    # accept the proposal or not, with the appropriate probability
    if(rbinom(1, 1, r) == 1) {
        theta_posterior_sample$mu[i] <- mu_proposal
        theta_posterior_sample$log_sigma_sq[i] <- log_sigma_sq_proposal
        theta_posterior_sample$sigma_sq[i] <- exp(log_sigma_sq_proposal)
    } else {
        theta_posterior_sample$mu[i] <- previous_mu
        theta_posterior_sample$log_sigma_sq[i] <- previous_log_sigma_sq
        theta_posterior_sample$sigma_sq[i] <- exp(previous_log_sigma_sq)
    }
}

# discard burn-in
theta_posterior_sample <- theta_posterior_sample[-seq_len(burn_in), ]
```

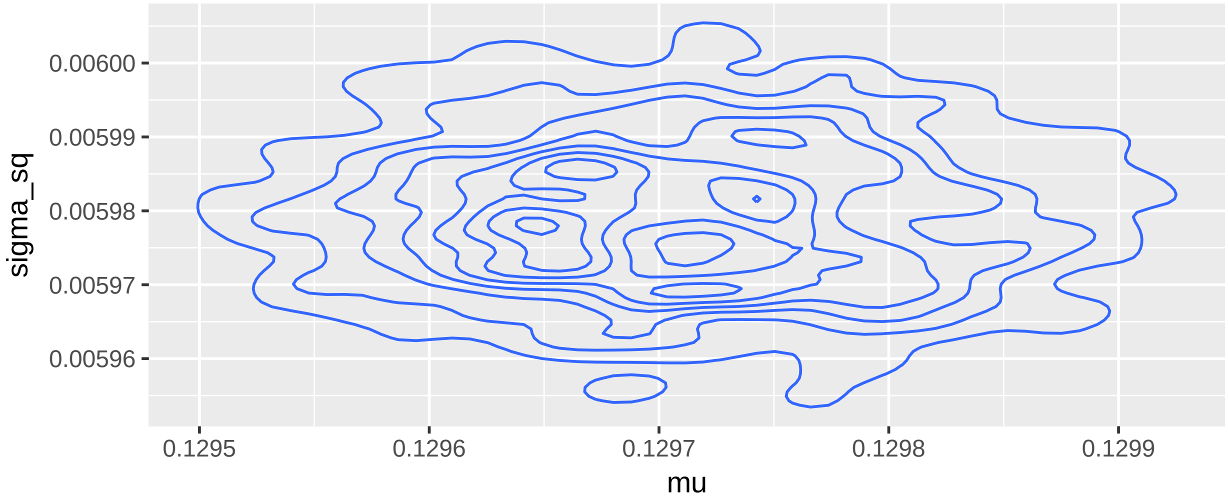
Plots to represent the approximate joint posterior distribution of μ, σ^2

Each point in the plot is a sample from the joint posterior of $\mu, \sigma^2 | x_1, \ldots, x_n$.

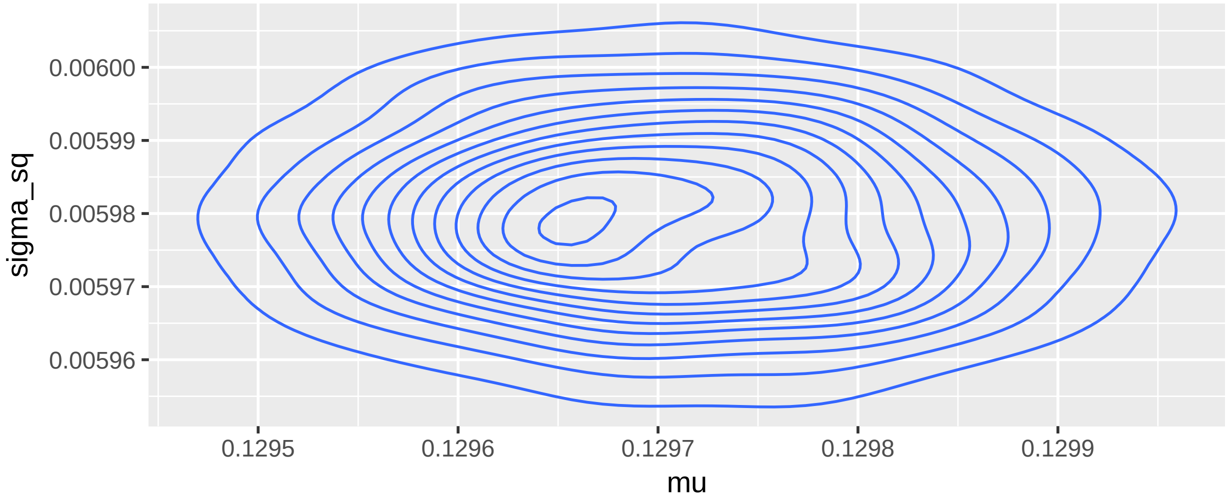
```r
ggplot(data = theta_posterior_sample, mapping = aes(x = mu, y = sigma_sq)) +
  geom_point(alpha = 0.4)
```



```
ggplot(data = theta_posterior_sample, mapping = aes(x = mu, y = sigma_sq)) +
  geom_density2d()
```



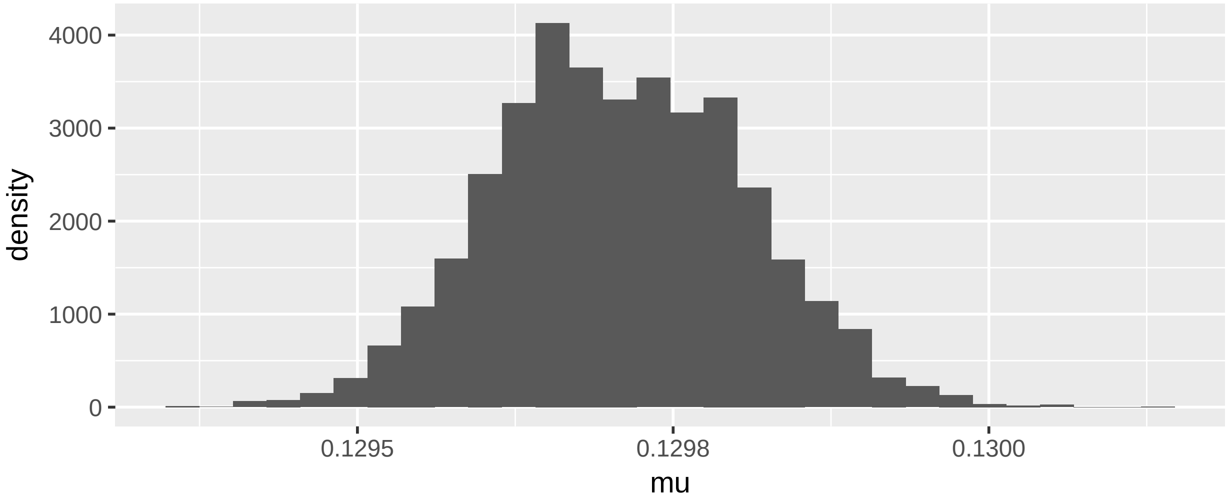
```
ggplot(data = theta_posterior_sample, mapping = aes(x = mu, y = sigma_sq)) +
  geom_density2d(h = c(0.00015, 0.000015))
```



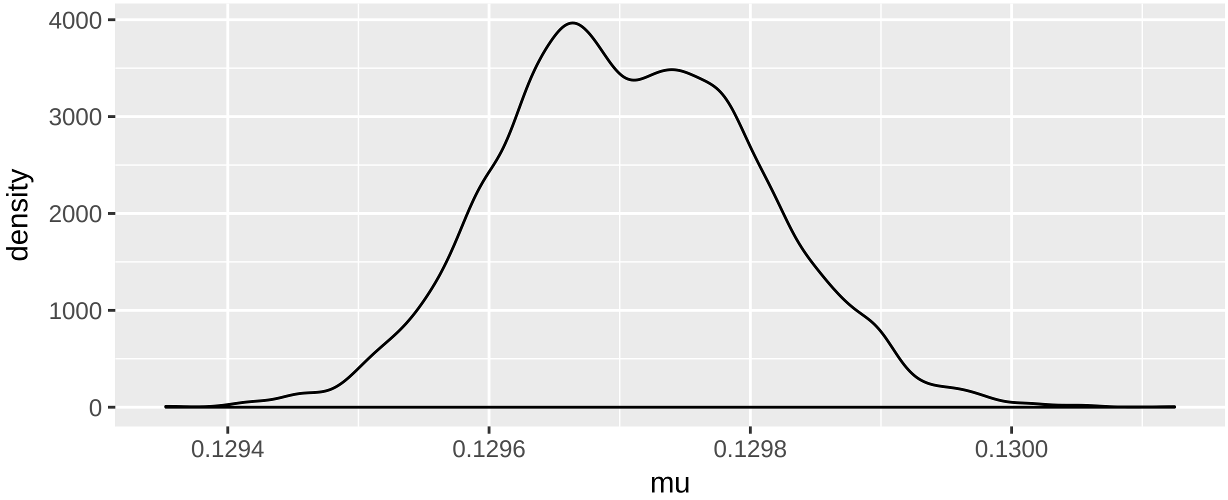
Plots to represent the approximate marginal posterior distribution of μ

```
ggplot() +
  geom_histogram(data = theta_posterior_sample, mapping = aes(x = mu, y = ..density..))
```

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



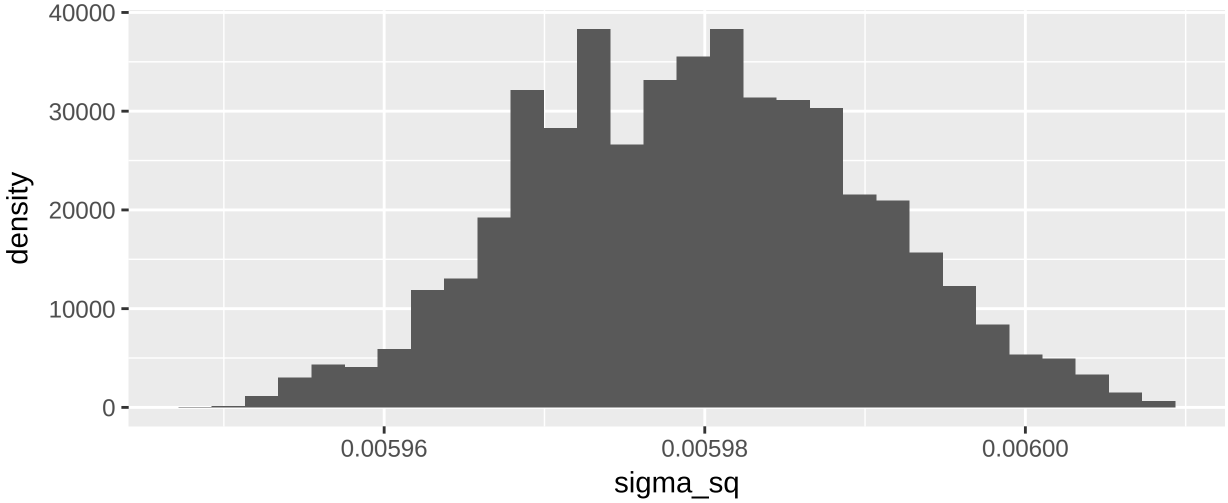
```
ggplot() +
  geom_density(data = theta_posterior_sample, mapping = aes(x = mu))
```



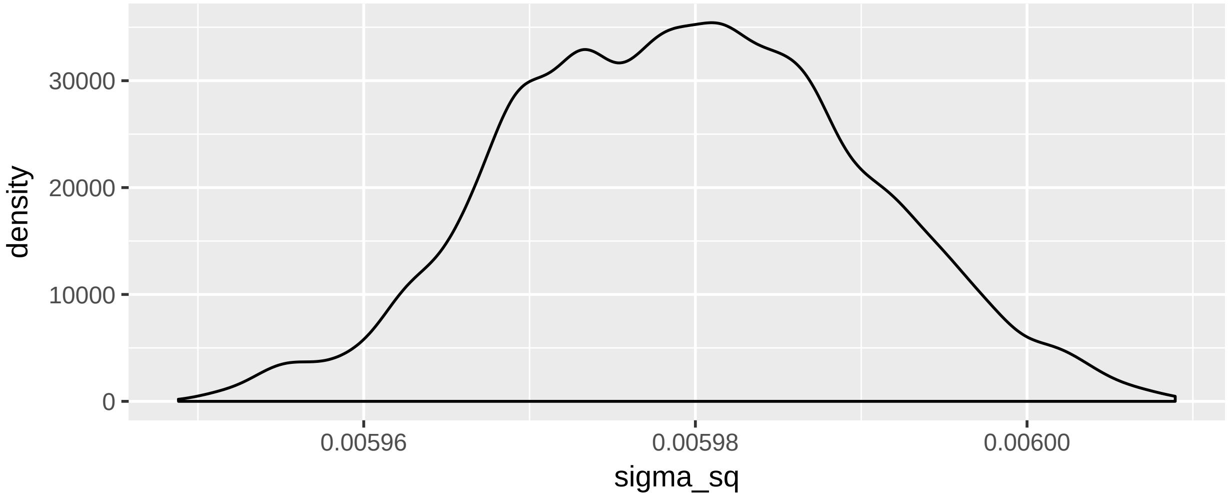
Plots to represent the approximate marginal posterior distribution of σ^2

```
ggplot() +
  geom_histogram(data = theta_posterior_sample, mapping = aes(x = sigma_sq, y = ..density..))
```

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



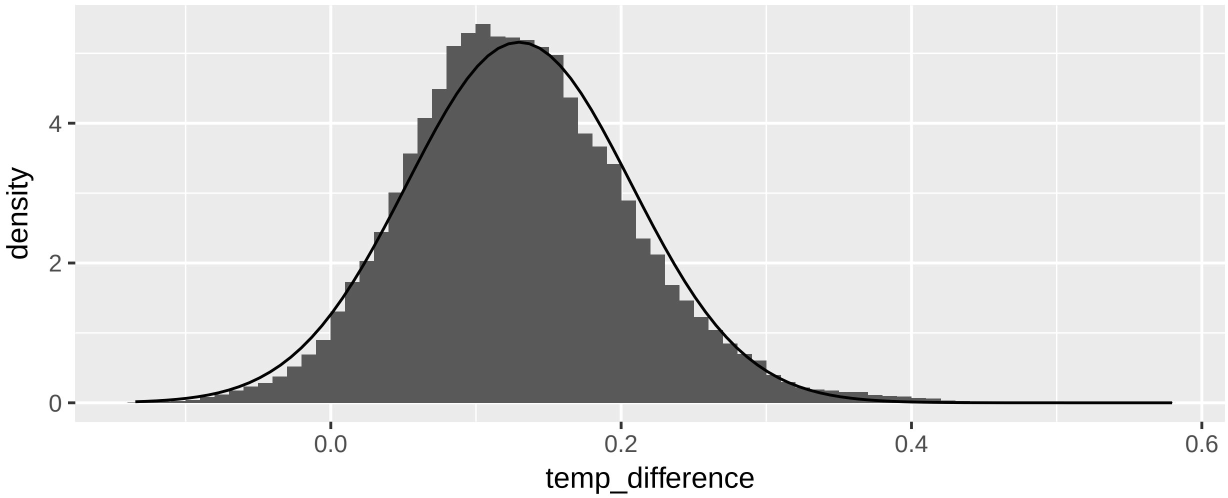
```
ggplot() +
  geom_density(data = theta_posterior_sample, mapping = aes(x = sigma_sq))
```



How does the model fit?

Compare a histogram of the data to the normal density with parameters set equal to the estimated posterior mean for μ and for σ^2.

```
ggplot(data = cmb, mapping = aes(x = temp_difference)) +
  geom_histogram(center = 0.005, binwidth = 0.01, mapping = aes(y = ..density..)) +
  stat_function(fun = dnorm,
    args = list(mean = mean(theta_posterior_sample$mu), sd = sqrt(mean(theta_posterior_sample$sigma_sq))))
```


density
0
2
4
0.0
0.2
0.4
0.6
temp_difference