Week 6 Module

Steps of Bayesian Data Analysis

For a Bayesian analysis we will follow these steps:

1. **Identify the data relevant to the research questions.** What are the measurement scales of the data? Which data variables are to be predicted, and which data variables are supposed to act as predictors?

2. **Define a descriptive model for the relevant data.** The mathematical form and its parameters should be meaningful and appropriate to the theoretical purposes of the analysis.

3. **Specify a prior distribution on the parameters.** The prior must pass muster with the audience of the analysis, such as skeptical scientists.

4. **Use Bayesian inference to re-allocate credibility across parameter values.** Interpret the posterior distribution with respect to theoretically meaningful issues (assuming that the model is a reasonable description of the data; see next step).

5. **Check that the posterior predictions mimic the data with reasonable accuracy (i.e., conduct a 'posterior predictive check').** If not, then consider a different descriptive model.

Example of Bayesian Analysis on a binary outcome

Consider estimating the probability that a house in Seattle has more than 2 bathrooms.

1. Identify relevant data

```
seattle <- read_csv('http://www.math.montana.edu/ahoegh/teaching/stat408/datasets/SeattleHousing.csv')
  mutate(more_than2baths = bathrooms > 2)

z <- sum(seattle$more_than2baths)
N <- nrow(seattle)
```

2. Define a descriptive statistical model for the relevant data.

A descriptive model denoted as $p(\mathcal{Y}|\theta)$, where $\mathcal{Y} = \{y_1, y_2, \ldots, y_n\}$ the data for binary outcomes of rolling the die such that $y_i = 1$ if a house has more than 2 bathrooms and $y_i = 0$ otherwise.

Here we use a binomial distribution as the statistical model for our data, which can be written as:

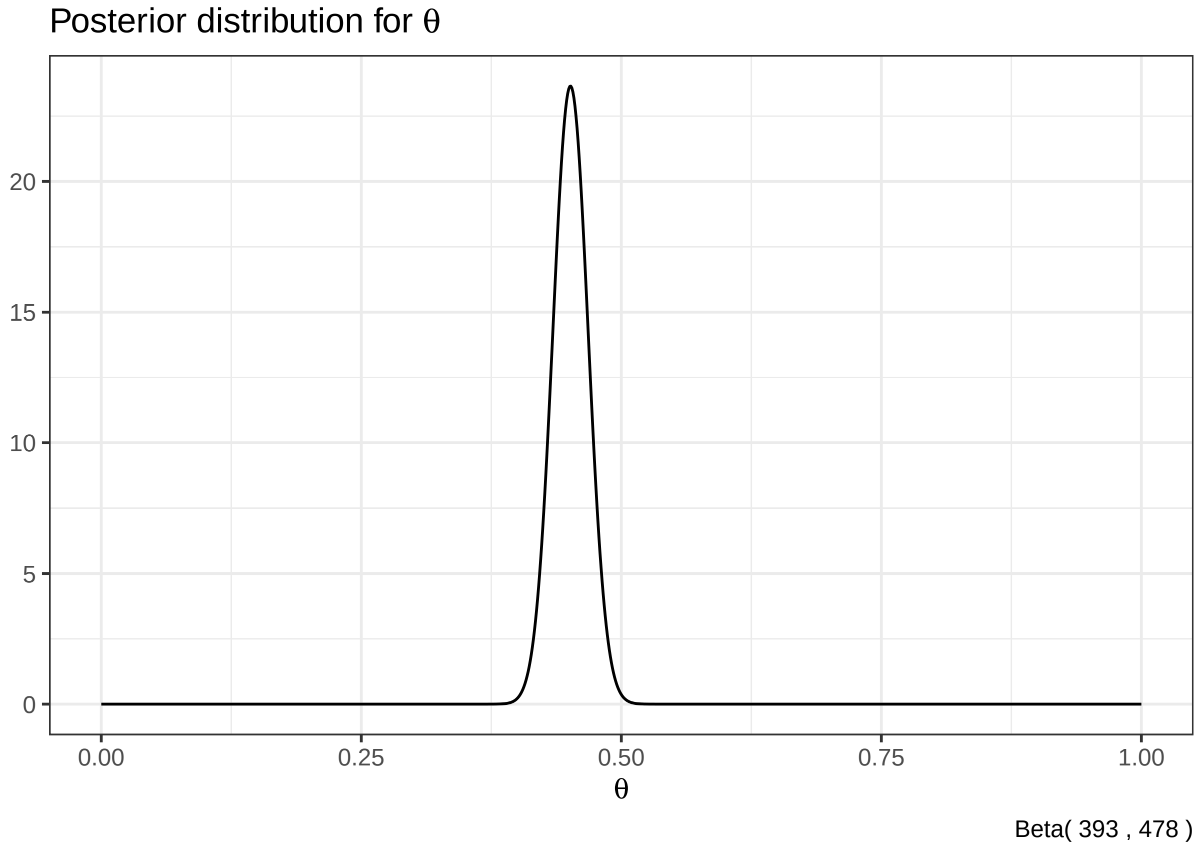
3. Specify a prior distribution on the parameters

Given the probability distribution in part 2, our parameter of interest is θ (the probability that a house has more than 2 bathrooms).

4. Use Bayesian inference to re-allocate credibility across parameter values.

We have seen that given binomial data and a beta prior on θ we have an exact posterior distribution for θ

$$
\begin{aligned}
p(\theta|z,N) &= \frac{p(z,N|\theta)p(\theta)}{p(z,N)} \\
&= \frac{p(z,N|\theta)p(\theta)}{\int p(z,N|\theta)p(\theta)d\theta} \\
&= \frac{\theta^z(1-\theta)^{(N-z)} \times (\Gamma(a)\Gamma(b)/\Gamma(a+b))\theta^{(a-1)}(1-\theta)^{(b-1)}}{\int \theta^z(1-\theta)^{(N-z)} \times (\Gamma(a)\Gamma(b)/\Gamma(a+b))\theta^{(a-1)}(1-\theta)^{(b-1)}d\theta} \\
&= \frac{(\Gamma(a)\Gamma(b)/\Gamma(a+b))\theta^{z+a-1}(1-\theta)^{b+N-z-1}}{(\Gamma(a)\Gamma(b)/\Gamma(a+b))\int \theta^{z+a-1}(1-\theta)^{b+N-z-1}d\theta} \\
&= \frac{\theta^{z+a-1}(1-\theta)^{b+N-z-1}}{\int \theta^{z+a-1}(1-\theta)^{b+N-z-1}d\theta} \\
&= \frac{\theta^{z+a-1}(1-\theta)^{b+N-z-1}}{\Gamma(a+b+N)/(\Gamma(a+z)\Gamma(N-z+b))} \\
&= \frac{(\Gamma(a+z)\Gamma(N-z+b))\theta^{z+a-1}(1-\theta)^{b+N-z-1}}{\Gamma(a+b+N)} \\
&\sim Beta(a+z, b+N-z)
\end{aligned}
$$



5. Check that the posterior predictions mimic the data with reasonable accuracy (i.e., conduct a 'posterior predictive check').

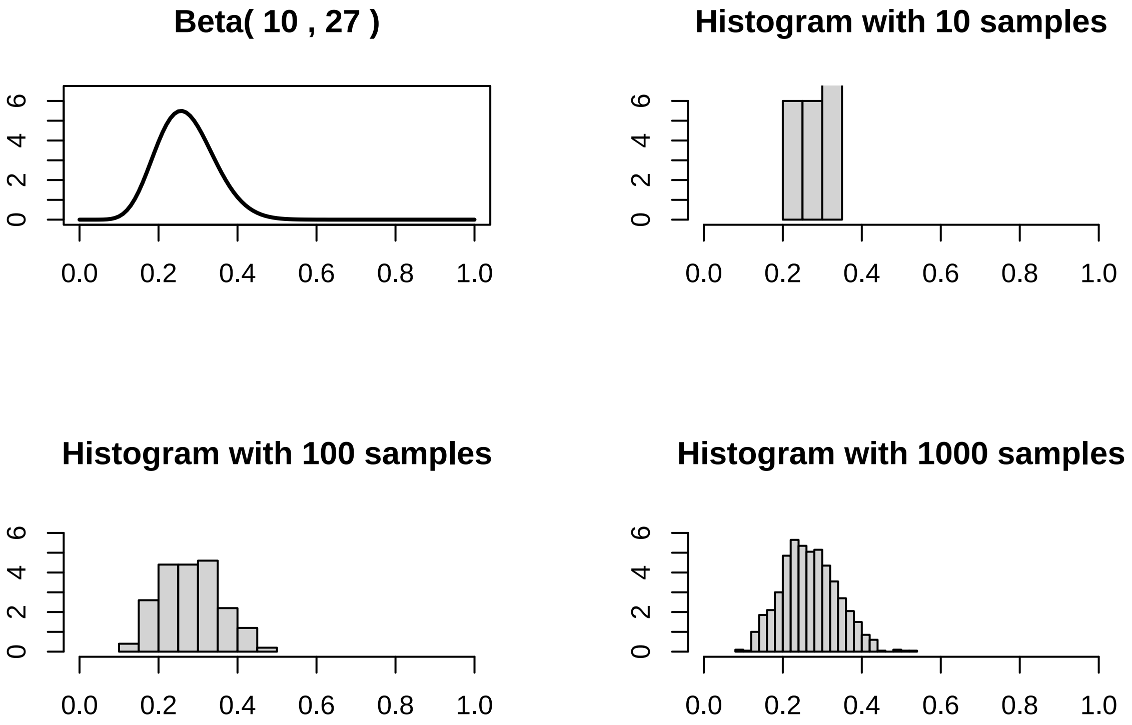
We will come back to this later in the semester

Approximating a Distribution with a Large Sample

We saw that a beta distribution was a conjugate prior for a sampling model, meaning that the posterior was also a beta distribution. This prior specification allows easy posterior computations because the integration in the denominator of Bayes rule $\int p(y|\theta)p(\theta)d\theta$ can be analytically computed.

In many situations, this type of prior is not available and we need to use other means to understand the posterior distribution $p(\theta|y)$. MCMC is a tool for taking samples from the posterior when $\int p(y|\theta)p(\theta)d\theta$ is messy and $p(\theta|y)$ does not have the form of a known distribution.

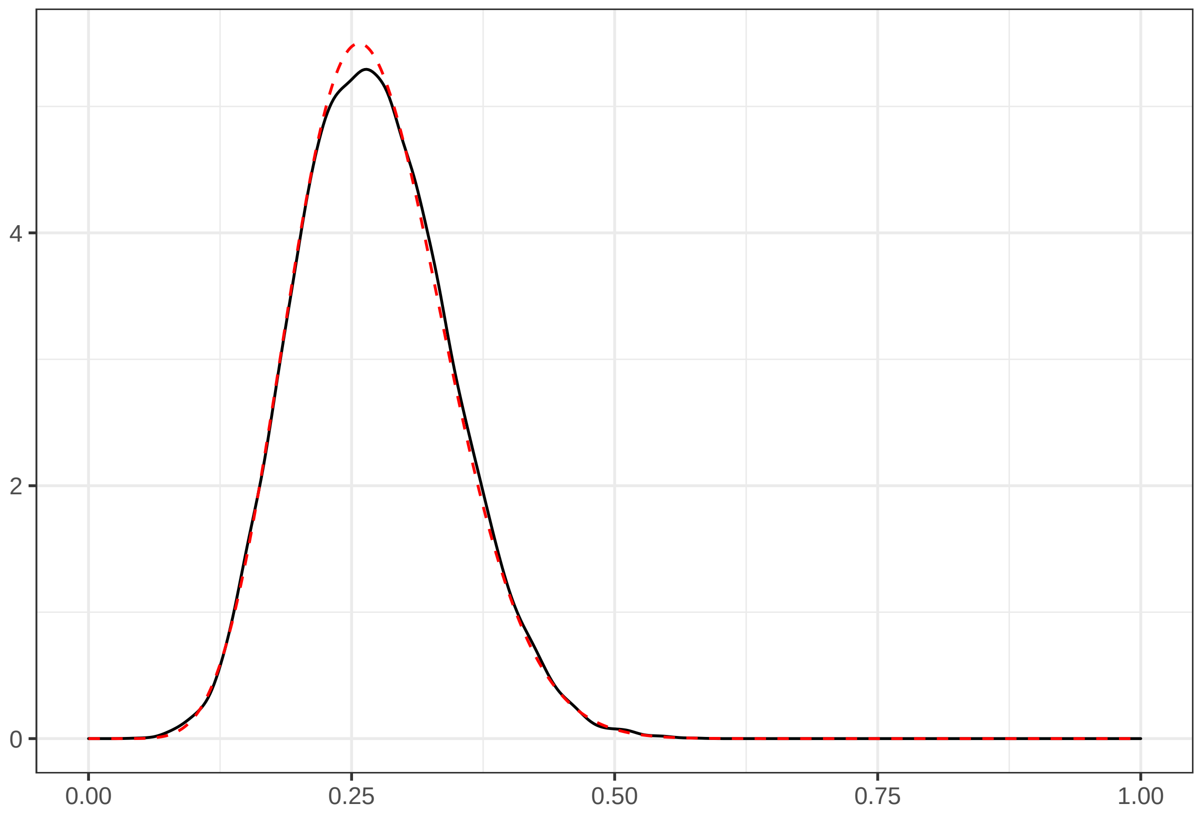
Previously we computed the mean and variance of distributions using Monte Carlo techniques. Now consider taking sample to visualize an entire distribution.



As the sample size gets larger, the histogram representation begins to look more like the true distribution. Additionally the moments of the sampled distribution approach that of the true distribution.

The true mean is $\frac{a}{a+b} = 0.27$ and quantiles can be computed using `qbeta(.025,a,b)` $= 0.142$ and `qbeta(.975,a,b)` $= 0.422$

- with 10 samples: the mean is 0.278 and the quantiles are (0.213, 0.346)
- with 100 samples: the mean is 0.281 and the quantiles are (0.162, 0.429)
- with 1000 samples: the mean is 0.268 and the quantiles are (0.141, 0.413)
- with 10000 samples: the mean is 0.271 and the quantiles are (0.144, 0.422)


4
2
0
0.00
0.25
0.50
0.75
1.00

Markov Chain Monte Carlo

For now, I'm going to hide details of the algorithm, but assume it is similar to what we've just done. There is a multipurpose software called JAGS that will allow us to fit Bayesian models in a variety of scenarios.

1. download JAGS
2. install `rjags` and `runjags`
3. JAGS references:

- https://bookdown.org/kevin_davisross/bayesian-reasoning-and-methods/introduction-to-jags.html
- https://people.stat.sc.edu/hansont/stat740/jags_user_manual.pdf

JAGS code will allow, or force, us to explicitly write out our probability model and prior distributions.

```
model_string <- "model{
  # Likelihood
  z ~ dbinom(theta, N)

  # Prior
  theta ~ dbeta(alpha, beta)
  alpha <- 1 # prior successes
  beta <- 1 # prior failures
}"
```

```
dataList = list(z = z, N = N)

model <- jags.model(file = textConnection(model_string), data = dataList)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 1
##    Unobserved stochastic nodes: 1
##    Total graph size: 4
##
## Initializing model
```

```
update(model, n.iter = 5000) # warmup


# take samples
posterior_sample <- coda.samples(model,
                       variable.names = c("theta"),
                       n.iter = 10000)

summary(posterior_sample)
```

```
##
## Iterations = 6001:16000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
```

```
##           Mean             SD       Naive SE Time-series SE
##       0.451091       0.016998       0.000170       0.000216
##
## 2. Quantiles for each variable:
##
##   2.5%    25%    50%    75%  97.5%
## 0.4182 0.4392 0.4510 0.4626 0.4849
```

With the true posterior we get

Example of Bayesian Analysis on a continuous outcome

Recall Q3 from Lab 1,

> Now let's consider estimating the temperature at Hyalite Canyon (at the reservoir) at 10 AM on Saturdays in February.

Let's conduct the first 4 steps of the Bayesian analysis

1. Identify the data relevant to the research question.

We have access to historical weather data from a snotel site located near Hyalite Canyon, https://wcc.sc.egov.usda.gov/nwcc/site?sitenum=754. In particular, assume we've scraped data from Feb 15th going back to 1989. These are average temperatures (difference in Max - Min), so might require restating the research question.

```
feb15_temp <- c(1, -6.3,  27.3, 21.4, -13.2, 27, 8.8, 23.5, 26.6, 16, 22.5, 19.2,
  18, 26.1, 20.7, 8.4, 2.3, 27.1, 21.9, 16.2, 27.1, 34.5, 13.5, 22.8, 28.2,
  25, 31.6, 34.2, 12.7, 26.2, 18.3, 19.6, 23.2, 2.3)
```

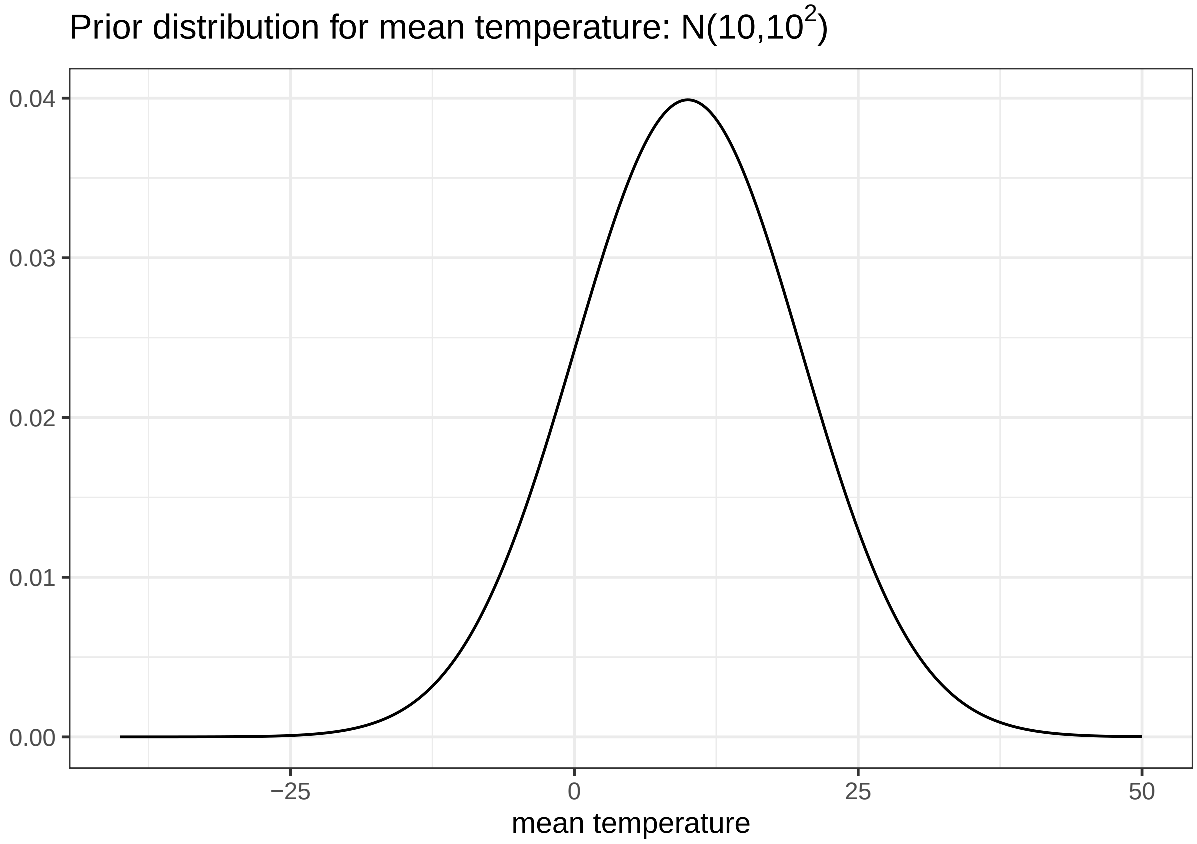
2. Identify a descriptive statistical model for the relevant data. Then interpret the statistical parameters in that model.

A normal distribution seems reasonable to use in this case. Temperature can go below zero, so a distribution restricting responses to positive values is not necessary. The normal distribution, commonly expressed through the idea of a bell curve, has two parameters: a mean and variance (or standard deviation). The mean temperature will give us the average temperature and Hyalite in February; whereas, the standard deviation will encode how much variability is present in temperatures.

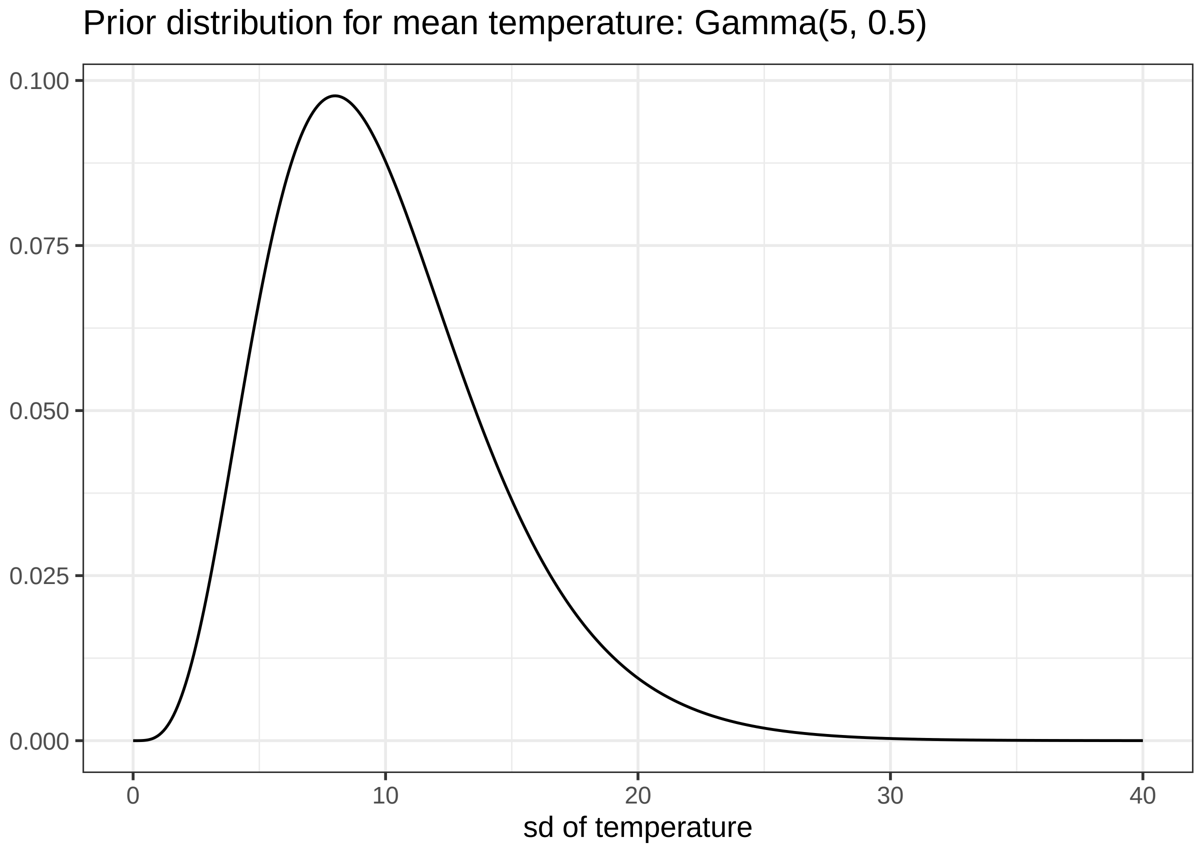
3. Specify a prior distribution for all parameters in the model.

Given that we are using the normal distribution, we need to consider two prior distributions: one for the mean and one for the standard deviation (or variance),

```r
temp_seq <- seq(-40,50, length.out = 500)
tibble(vals = c(dnorm(temp_seq, 10, 10)),
       temperature = temp_seq) %>%
  ggplot(aes(x = temperature, y = vals)) +
  geom_line() + theme_bw() +
  ggtitle(expression(paste('Prior distribution for mean temperature: N(10,',10^2,')'))) +
  xlab('mean temperature') + ylab('')
```



```r
sd_seq <- seq(0,40, length.out = 500)
tibble(vals = c(dgamma(sd_seq, 5, .5)),
       temperature = sd_seq) %>%
  ggplot(aes(x = temperature, y = vals)) +
  geom_line() + theme_bw() +
  ggtitle(expression(paste('Prior distribution for mean temperature: Gamma(5, 0.5)'))) +
  xlab('sd of temperature') + ylab('')
```



4. Use Bayesian inference to re-allocate credibility across parameter values.

Unfortunately, the calculus is not as straightforward this time around…

$$\begin{aligned}
p(\mu, \sigma|\mathcal{Y}) &= \frac{p(\mathcal{Y}|\mu, \sigma)p(\mu)p(\sigma)}{p(\mathcal{Y})} \\
&\approx \frac{N(\mathcal{Y}|\mu, \sigma)N(\mu), Gamma(\sigma)}{\int N(\mathcal{Y}|\mu, \sigma)N(\mu), Gamma(\sigma)d\mu\ d\sigma}
\end{aligned}$$

```
model_normal<- "model{
  # Likelihood
  for (i in 1:n){
    y[i] ~ dnorm(mu, 1/sigma^2)
  }

  # Prior
  mu ~ dnorm(10, 1/7^2)
  sigma ~ dgamma(5, .5)
}"
```

```
dataList = list(y = feb15_temp, n = length(feb15_temp))

model <- jags.model(file = textConnection(model_normal), data = dataList)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 34
##    Unobserved stochastic nodes: 2
##    Total graph size: 47
##
## Initializing model
```

```r
update(model, n.iter = 5000) # warmup


# take samples
posterior_sample <- coda.samples(model,
                       variable.names = c("mu", 'sigma'),
                       n.iter = 10000)

summary(posterior_sample)
```

```
##
## Iterations = 6001:16000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##        Mean    SD Naive SE Time-series SE
## mu    18.05 1.860  0.01860        0.01860
## sigma 11.28 1.357  0.01357        0.01841
##
## 2. Quantiles for each variable:
##
##        2.5%   25%   50%   75% 97.5%
## mu    14.43 16.78 18.03 19.27 21.66
## sigma  9.00 10.32 11.14 12.10 14.26
```

```r
summary(lm(feb15_temp ~ 1))
```

```
##
## Call:
## lm(formula = feb15_temp ~ 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.838  -4.513   3.012   7.862  15.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   18.638      1.915   9.731 3.19e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 11.17 on 33 degrees of freedom
```