Project 3: The Normal-Normal Conjugate Family

Sam Albertson, Suz Angermeier, and Dani Vaithilingam

12/16/2023

Contents

1 Background

This template uses a similar format as the course notes for this class. In fact, you should imagine that you are designing a lesson to teach the normal-normal conjugate model as you complete this template. Any good lesson needs an example data set to illustrate the concepts. Feel free to use something from a project you've been working on (be careful not to use data without permission!), find a dataset online, or create a fictitious example and simulate the data for the example.

In this section you should give some brief background on the kinds of problems that the normal-normal model is appropriate for, what the target of inference is, and briefly introduce the example you are going to use.

The normal-normal Bayesian model is a useful framework for the analysis of continuous data, particularly when dealing with parameters that can take on a wide range of values without strict upper or lower bounds. This family of Bayesian models uses a Gaussian distribution to represent both the prior and likelihood functions. Our target of inference is μ, the mean parameter of a Gaussian distribution.

For this example, we are investigating whether feeding chicks a protein-rich diet affects their final weight compared to the expectation of weight for chicks fed with a non-protein-rich diet. Our dataset is modified from ChickWeight (https://www.rdocumentation.org/packages/datasets/versions/3.6.2/topics/ChickWeight) from R.

Our specific objective is to quantify the impact of a protein-rich diet on the expected weight (μ) of chicks, leveraging Bayesian inference with a normal-normal model. This approach allows us to incorporate prior beliefs about μ (the mean weight under a non-protein-rich diet) and update these beliefs based on observed data to estimate the effect of the protein-rich feed.

Next, we'll delve into setting up the model and conducting Bayesian analysis to address this question effectively.

2 Normal PDF

Imagine you are explaining the normal distribution to someone who has never heard of it. Include all the basics such as the equation for the PDF, the mean, median, mode, variance, etc. as well as a plain language description of what kinds of problems normal distributions are good models for. Give an example use of a normal distribution and how to interpret it in the context of your example (this would be the start of the running example for the rest of the lesson based on the dataset you found or simulated as described above).

2.0.1 Basics of the Normal Distribution

The normal distribution, also referred to as the Gaussian distribution, is the most common distribution used in statistical modeling. It is characterized by its bell-shaped curve, which is symmetric around its mean (μ).

In cases where we know the mean and typical variance in our data, and where we expect the data we sample to be symmetrically distributed on either side of the mean, many different random processes result in a normal sampling distribution. The symmetry is convenient, because if we can specify the mean of the distribution, we know that will also be equivalent to both the median and the mode. In other words, mean, median, and mode of a normal distribution are all equal and located at μ.

Also convenient is the fact that the mean and variance are defined independently of each other, allowing us to freely adjust the shape and location of our distribution as needed. The variance controls the spread of the distribution, with a larger variance meaning the data points are further from the mean and a smaller variance meaning they are clustered closer to the mean.

Given a mean μ and variance σ^2 (or a standard deviation of σ), we can specify the PDF as:

$pdf(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$

We can shorthand this notationally for a normally-distributed random variable $Y \sim N(\mu, \sigma^2)$.

2.0.2 Applications of the Normal Distribution

Normal distributions are useful for modeling the distribution of data which is continuous, unbounded, and symmetrically distributed around its mean, which leads to it being used to represent most continuous variables in statistics. Common examples of data that can be modeled using the normal distribution include heights of people, measurement errors, test scores, and natural phenomena like rainfall amounts.

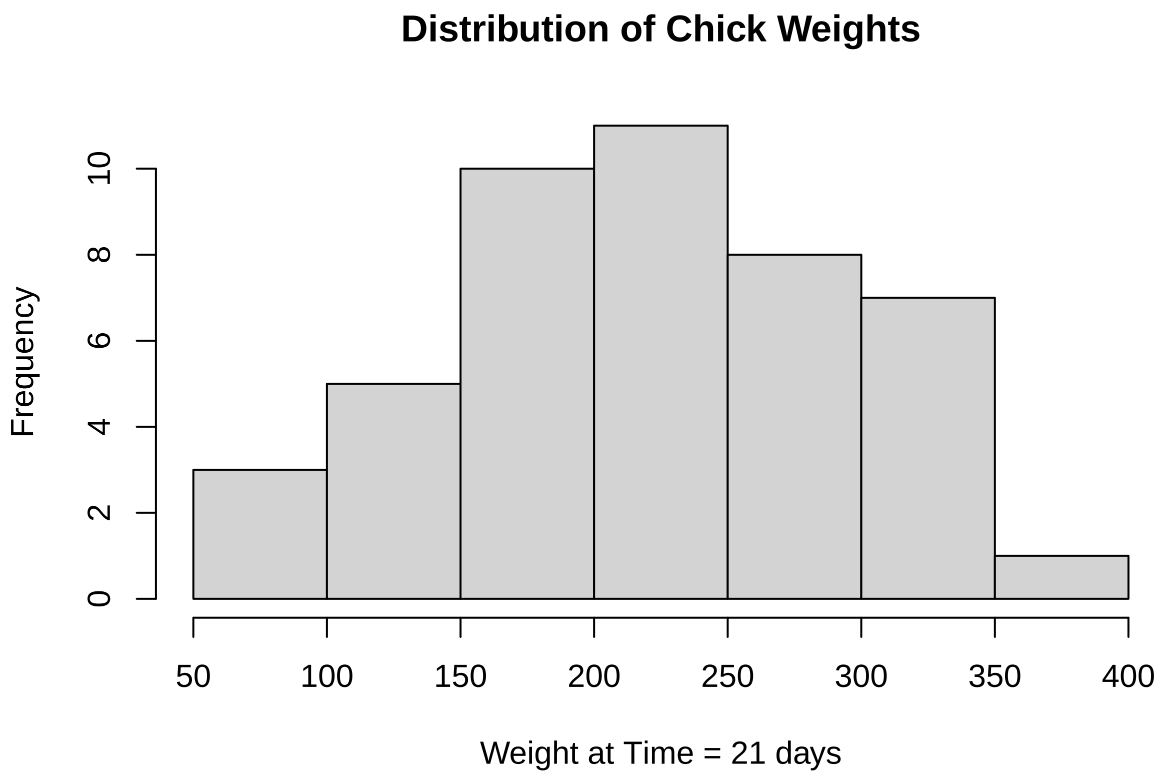
2.0.3 Example Use of Normal Distribution

In the context of the ChickWeight dataset, suppose we are interested in modeling the final weights of chicks after being fed a protein-rich diet. Let's say the final weights follow a normal distribution with a mean $\mu = 219$ and a standard deviation $\sigma = 71.5$.

In standard notation, this is written as Final Weight $\sim N(219, 30^2)$.

This implies that most chicks' final weights will be around 219 grams, with some variation around the mean due to individual differences and other factors, as shown in the histogram below.

```r
hist(subset(ChickWeight, Time == 21)$weight, main = "Distribution of Chick Weights", xlab = "Weight at
```



Understanding and applying the normal distribution allows us to make probabilistic statements about chick weights and assess the impact of different diets on their growth. In the following sections, we'll explore how Bayesian inference with a normal-normal model can provide insights into the effect of protein-rich feed on chick weights using the principles of the normal distribution.

3 Normal Prior

Discuss how the normal distribution can be used to describe information about the target parameter prior to conducting a study. After explaining this in general, then relate it to the specific example you are using in this lesson.

3.0.1 The role of the prior in Bayesian inference

A prior distribution allows us to incorporate our beliefs and prior knowledge about the parameter of interest. It influences the posterior distribution (our updated belief about μ after observing data) through Bayesian updating and helps balance the influence of observed data with prior beliefs. We select what type of distribution (normal, beta, etc) we want to use for our prior based on the range of plausible values the parameter of interest can take.

The normal distribution is one type of prior distribution that can be used to describe what we know or assume about our target parameter, μ, based on prior knowledge, previous studies or expert opinion before observing any of our data. The prior mean, μ_0, represents our best guess or expectation of the parameter before seeing our data. The prior variance, σ_0^2, reflects the uncertainty of our expectations about the parameter μ. A larger variance implies that we are more uncertain about the value of μ or that there is a wider range of plausible values for it. A smaller variance implies that we are more confident in our prior estimate for μ.

In our example, we are interested in finding the mean weight μ of the chicks in our dataset via a Bayesian analysis. Before we can estimate this with our dataset, we need to define a prior to represent the state of our existing knowledge. This prior will be defined using the normal distribution, which means we need to specify both our expected mean and our standard deviation.

In the ChickWeight example:

In our ChickWeight example, we aim to estimate the mean weight (μ) of chicks fed a protein-rich diet using Bayesian analysis. Before examining the dataset, we establish a prior distribution for μ:

- **Prior Mean (μ_0):** Based on our research (https://www.agrifarming.in/poultry-feed-chart-and-weight-chart#layer-poultry-feed-chart-and-weight-chart), the mean body weight for layer poultry at three weeks old is around 190 grams. Hence, we set $\mu_0 = 190$ grams.
- **Prior Mean (σ_0^2):** We use a weak prior of 47.5 grams as our standard deviation, which is approximately 1/4th of the mean. This indicates uncertainty in our prior belief.

By defining our prior as $\mu \sim N(190, 47.5^2)$, we are saying that, before observing any data, we believe that μ is around 190 grams, but we are not certain of this estimate.

As we observe the chick weights, Bayesian updating will adjust our belief about μ from the prior distribution to the posterior distribution, reflecting our updated understanding based on both prior knowledge and observed data.

In the subsequent steps of our analysis, we'll combine this prior with the likelihood function (derived from the observed data) to derive the posterior distribution of μ, which will guide our inference about the effect of protein-rich feed on chick weights. The strength of the prior, relative to the observed data, will determine the degree to which our prior beliefs influence our final conclusions.

4 Normal Likelihood

Explain what a likelihood function is in general and show the equation for the normal likelihood, explaining each part of the function in plain English. Show the likelihood for your example data. Include a plot of the likelihood function and explain how to interpret it in plain English.

Show how the likelihood function is derived. Go slowly (don't do too many steps of the algebra simultaneously) and explain each step you're taking. In particular, make sure to add in the bit that your textbook skips in Equation 5.13.

A likelihood function is the probability density of the observed data viewed as a function of the parameters of a statistical model. In simpler terms the likelihood function is the probability of observing data x assuming

θ is the actual parameter. The maximum value in the distribution is the maximum likelihood estimate is the point estimate for θ of the likelihood function.

Our normal likelihood distribution function of μ is found from $L(\mu|\vec{y} = f(\vec{y}|\mu)$. Now assume that σ is a known constant, we can simplify that the likelihood up to a proportionality constant by dropping the terms that don't depend upon μ. So for a $\mu \in (-\infty, \infty)$, we have that the likelihood is calculated from:

$L(\mu|\vec{y}) \propto \prod_{i=1}^{n} \exp\left[-\frac{(y_i-\mu)^2}{2\sigma^2}\right] = \exp\left[-\sum_{i=1}^{n} \frac{(y_i-\mu)^2}{2\sigma^2}\right]$

This can be simplified down, starting from $\exp\left[-\sum_{i=1}^{n} \frac{(y_i-\mu)^2}{2\sigma^2}\right]$.

$$L(\mu|\vec{y}) \propto \exp\left[-\sum_{i=1}^{n} \frac{(y_i-\mu)^2}{2\sigma^2}\right]$$

First we take to log of the formula to remove the exponents

$$ln(L(\mu|\vec{y})) \propto ln(\exp\left[-\sum_{i=1}^{n} \frac{(y_i-\mu)^2}{2\sigma^2}\right])$$

$$= -\sum_{i=1}^{n} \frac{(y_i-\mu)^2}{2\sigma^2}$$

Next we separate the fraction so that it is easier to deal with the summation value

$$= -\frac{1}{2\sigma^2}(\sum_{i=1}^{n}(y_i-\mu)^2)$$

Next we break down the summation variable into similar terms, by squaring the values

$$= -\frac{1}{2\sigma^2}(\sum_{i=1}^{n}(y_i^2 - 2y_i\mu + \mu^2))$$

Then we distribute out our summation symbol, constants are pulled out of the summation and go to the front & when a summation is placed with a constant we multiple the constant by n.

$$= -\frac{1}{2\sigma^2}(\sum_{i=1}^{n}(y_i^2) - 2\mu\sum_{i=1}^{n}(y_i) + n(\mu^2))$$

This gives us

$$= -\frac{1}{2\sigma^2}(\sum_{i=1}^{n}(y_i^2) - 2\mu\bar{y}n + n\mu^2)$$

The we multiply by $\frac{\frac{1}{n}}{\frac{1}{n}}$ this is equal to 1 so it does not effect the left side of the equation

$$= \frac{\frac{1}{n}}{\frac{1}{n}}(-\frac{1}{2\sigma^2}(\sum_{i=1}^{n}(y_i^2) - 2\mu\bar{y}n + n\mu^2))$$

$$= \frac{\frac{1}{n}}{\frac{1}{n}}(-\frac{\sum_{i=1}^{n}(y_i^2) - 2\mu\bar{y}n + n\mu^2)}{2\sigma^2})$$

$$= -\frac{\frac{1}{n}\sum_{i=1}^{n}(y_i^2) - 2\mu\bar{y}n + n\mu^2)}{2\sigma^2/n}$$

We reduce this to

$$= -\frac{(\bar{y}^2 - 2\mu\bar{y} + \mu^2)}{2\sigma^2/n}$$

Then we simplify the numerator to

$$-\frac{(\bar{y} - \mu)^2}{2\sigma^2/n}$$

Then we undo the $ln()$ by exponenting the equation on both sides and we are left with the likelihood function as shown in 5.13:

$$L(\mu|\vec{y}) \propto exp[-\frac{(\bar{y} - \mu)^2}{\frac{2\sigma^2}{n}}]\text{where } \mu \in (-\infty, \infty)$$

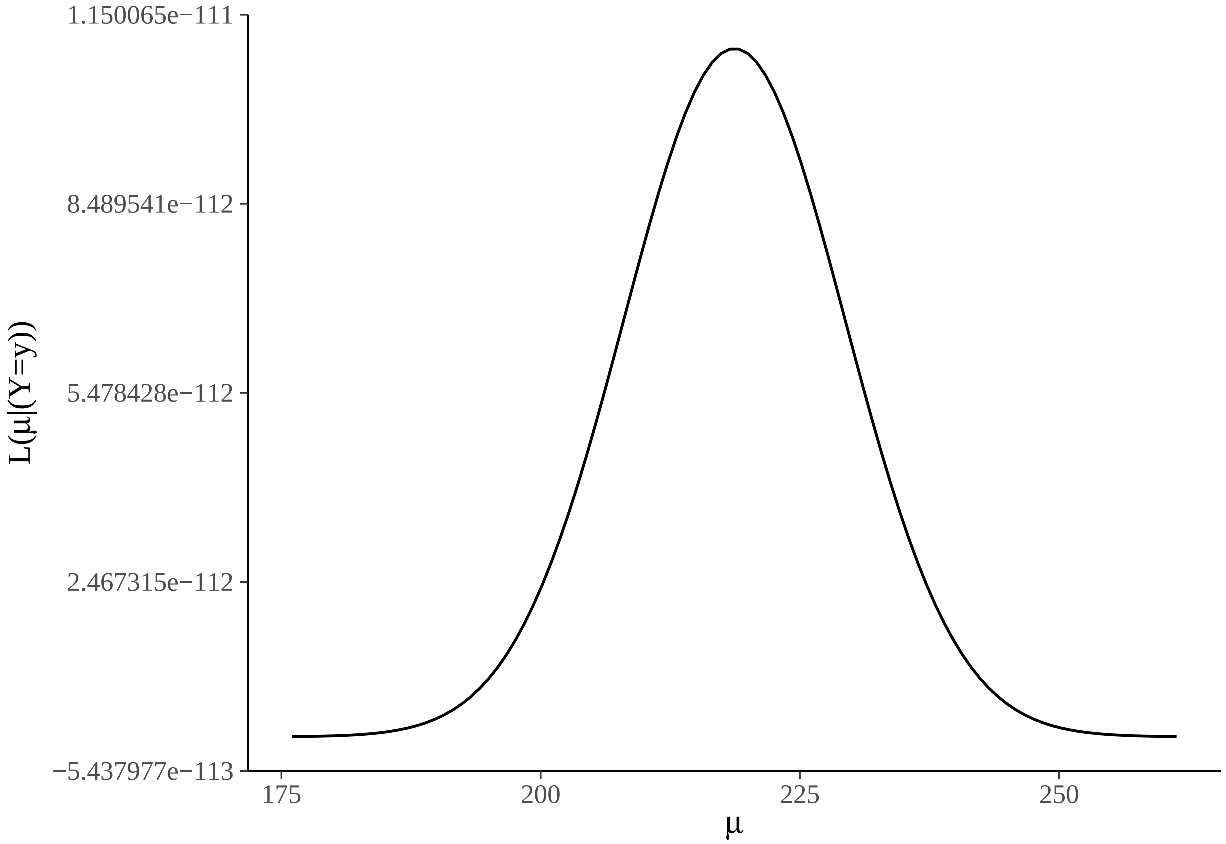
Now let us break this down into parts. The initial part, $L(\mu|\vec{y})$ is the likelihood function that represents the probability of observing the data $\vec{y}$ given the mean parameter μ. We have this probability proportional to our exponent values. The exponential values is the exponential of the squared difference between the sample mean and population mean divided by 2 times the population variance divided by our sample size. This tells how far apart the sample mean is from the population mean then we divide by a precision term, the larger the precision the better. This can be done from a value of μ from $(-\infty, \infty)$.

Now considering our data we have a likelihood function as follows, we have a population mean of 190 grams.

$$L(190|\text{End Weight}) \propto exp[-\frac{(\overline{\text{End Weight}} - 190)^2}{\frac{2*47.5^2}{45}}] \propto exp[-\frac{(\overline{\text{End Weight}} - 190)^2}{100.2778}]$$

Now to plot a likelihood function we have for our data with the population standard deviation:

```r
bayesrules::plot_normal_likelihood(data$EndWeight, sigma = 71.51027)
```



This plot represents the observed end weights and standard deviation in the likelihood distribution. We can see that mean of the end weight is more likely to between 215 grams and 225 grams. The most likely mean of the end weights is about 218 grams, but can easily be any of the values around 218 grams.

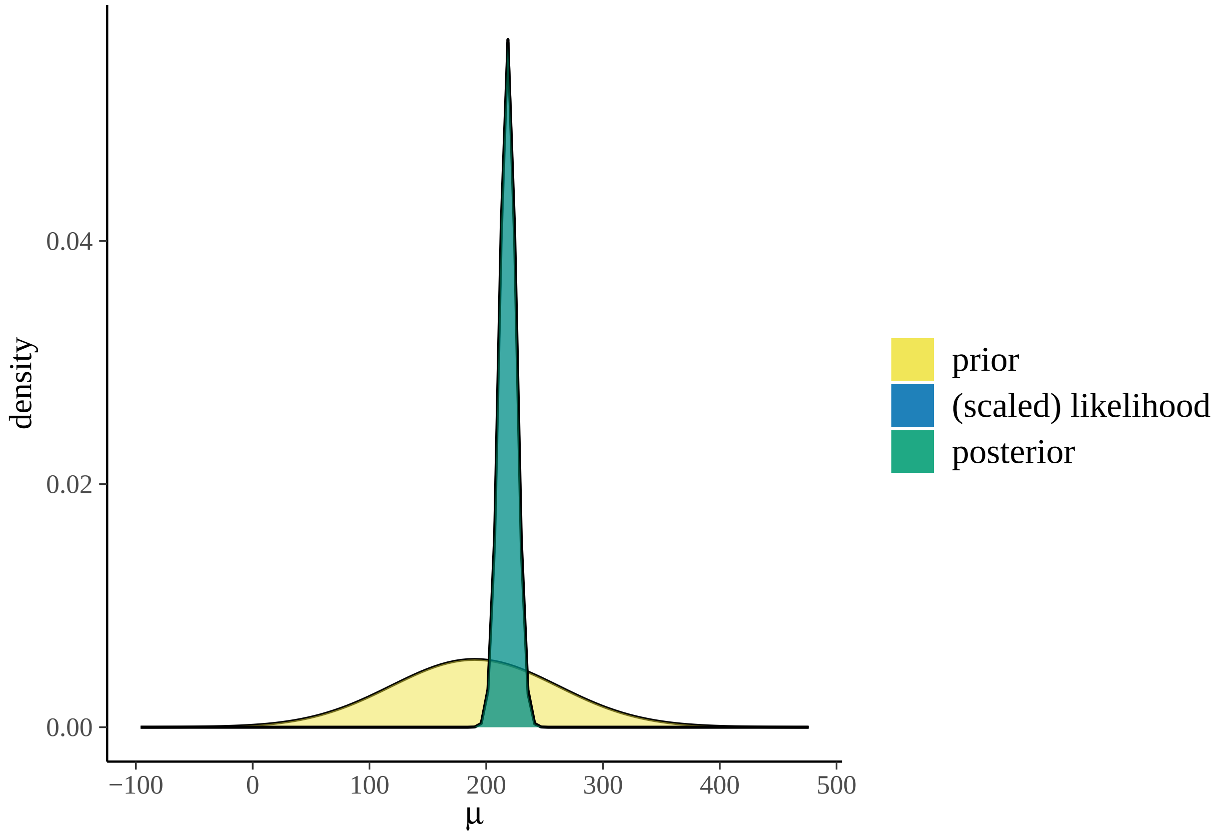
5 Normal-Normal Posterior

5.1 Balance of Prior Information and New Data

Design a graphical illustration of how prior information and new data are weighted differently in the normal-normal model.

The graphical illustration below demonstrates the balance between prior information and new data in the context of a normal-normal Bayesian model.

```
bayesrules::plot_normal_normal(mean = 190, sd = sd(data$EndWeight), sigma = 47.5,
                    y_bar = mean(data$EndWeight), n = 45)
```



5.1.1 Components of the Plot:

Prior Distribution:

The prior distribution, represented by the yellow curve, encapsulates our beliefs about the parameter μ (mean weight of chicks) before observing any data. In this example, the prior is specified with a mean of 190 grams and a standard deviation of 47.5 grams. The width of this curve reflects our uncertainty or variability in the prior belief.

Likelihood Distribution:

The likelihood distribution, depicted by the blue curve, is derived from the observed data. It represents how likely different values of μ are given the observed data (end weights of chicks). The likelihood is influenced heavily by the data and tends to be narrower compared to the prior, focusing more sharply around the observed data's central tendency.

Posterior Distribution:

The posterior distribution, shown in green, is the updated belief about μ after combining the prior and likelihood. It represents our current understanding of μ considering both prior knowledge and observed data. The posterior distribution is the result of Bayesian inference, where the influence of the prior and likelihood is balanced based on their respective uncertainties and information content.

5.1.2 Interpretations

When the prior distribution (yellow) is wide or non-informative, like in this case, it indicates that we have a lot of uncertainty about μ before observing the data. In cases like these, the liklihood distribution, blue, plays a more dominant role in shaping the posterior distribution.

If the prior were more narrow, the posterior would reflect more of a blend of the prior beliefs and the observed data.

The size of the dataset also influences the balance between the prior and the likelihood. Larger datasets provide more information and make the liklihood distribution more influential in determining the posterior.

5.2 Posterior Distribution for the Example

Show the posterior distribution, using equations and plots, for your example data set. Report a 95% equal tailed credible interval. Also report a posterior probability that is clinically relevant based on your example.

Posterior Distribution Equation:

$$\mu|\vec{y} \sim N(\theta\frac{\sigma^2}{n\tau^2+\sigma^2} + \bar{y}\frac{n\tau^2}{n\tau^2+\sigma^2}, \frac{\tau^2\sigma^2}{n\tau^2+\sigma^2})$$

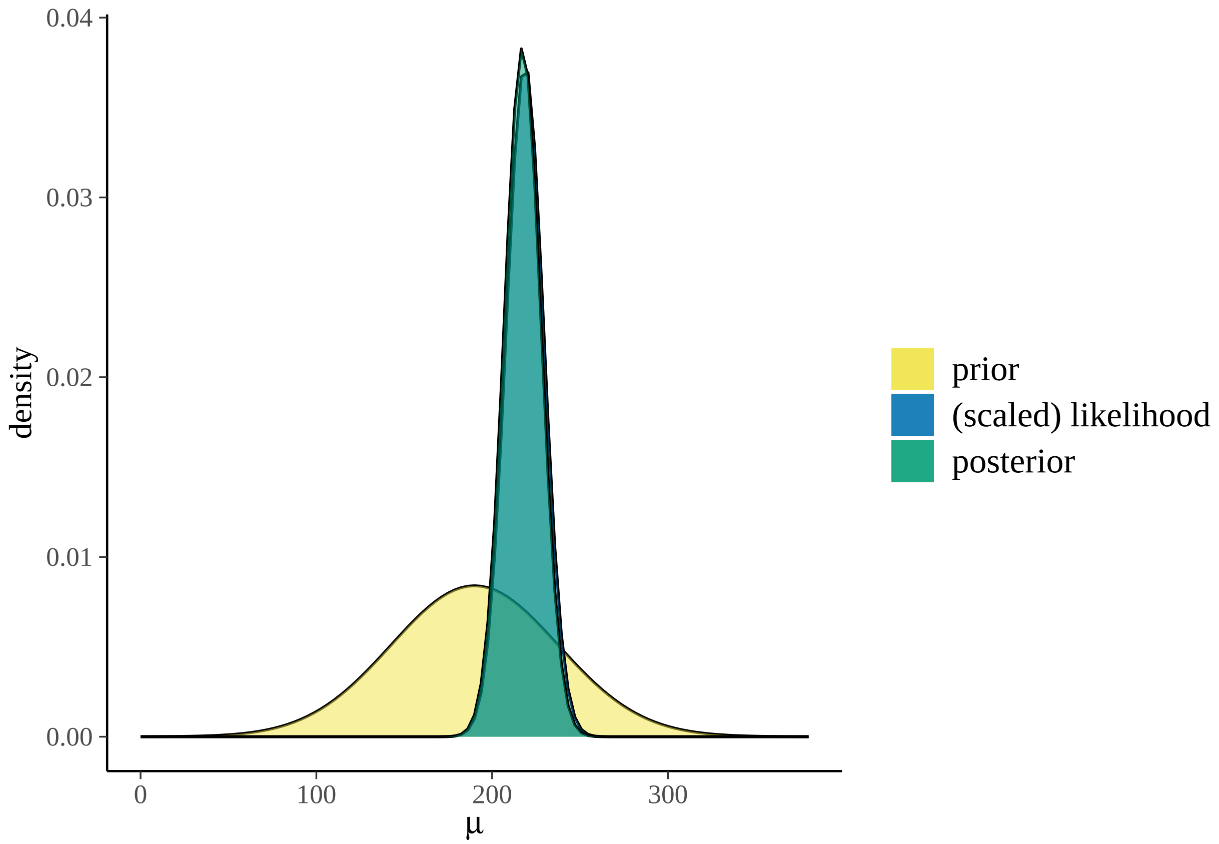
Thus for our data we have the equation equal to:

$$\mu|\vec{y} \sim N(190\frac{71.51027^2}{45(47.5^2)+71.51027^2} + 218.6889\frac{45(47.5^2)}{45(47.5^2)+71.51027^2}, \frac{47.5^2(71.51027^2)}{45(47.5^2)+71.51027^2})$$

$$\mu|\vec{y} \sim N(9.562797 + 208.2025, 108.1891)$$

$$\mu|\vec{y} \sim N(217.7653, 108.1891)$$

```
bayesrules::plot_normal_normal(mean = 190, sd = 47.5, sigma = 71.51027,
                               y_bar = 218.6889, n = 45)
```



```
bayesrules::summarize_normal_normal(mean = 190, sd = 47.5, sigma = 71.51027,
                                    y_bar = 218.6889, n = 45)
```

```
##       model     mean     mode       var      sd
## 1     prior 190.0000 190.0000 2256.2500 47.5000
## 2 posterior 217.3132 217.3132  108.1891 10.4014
```

```
pnorm(209, mean = 217.3132, sd = 10.4014, lower.tail = FALSE)
```

```
## [1] 0.787924
```

The 95% confidence interval is calculated as follows $217.3132 \pm 1.96(\frac{10.4014}{\sqrt{45}})$. So we have a 95% credible interval of our posterior distribution is $(214.274, 220.352)$.

The expected weight of a chick after 3 weeks (21 days) is 190 grams (https://www.agrifarming.in/poultry-feed-chart-and-weight-chart#layer-poultry-feed-chart-and-weight-chart). We want to know if protein rich feed increases the 3 week weight of the chicken by at least 10%.

Therefore we will consider the new diet to be successful if the chicks from our dataset have a greater than 80% probability of having an average weight of 209 grams or higher after 3 weeks (21 days) of being fed a high protein diet.

Now to understand if the new diet is successful we want to have an 80% probability of the end weight being 209 grams or above. Looking at our posterior distribution we have that 78.8% of our chicks have a weight of 209 grams or above. The new diet does not quite meet that standard, but is very close to being considered successful.

5.3 Derivation of the Posterior Distribution

EXTRA CREDIT: *Show the derivation of the posterior distribution for the normal-normal model. The textbook provides an example in Section 5.3.4. See if you can improve on the clarity of this illustration by showing each step in detail and explaining each step as you go.*

Let us go step-by-step through the derivation of the posterior distribution

$$f(\mu|\vec{y}) \propto f(\mu)L(\mu|\vec{y}) \propto exp[\frac{-(\mu-\theta)^2}{2\tau^2}]exp[-\frac{(\bar{y}-\mu)^2}{2\sigma^2/n}]$$

Next we expand the exponents out to get

$$f(\mu|\vec{y}) \propto f(\mu)L(\mu|\vec{y}) \propto exp[\frac{-\mu^2+2\mu\theta-\theta^2}{2\tau^2}]exp[\frac{-\mu^2+2\mu\bar{y}-\bar{y}^2}{2\sigma^2/n}]$$

We can combine these exp through algebra:

$$\propto exp[\frac{-\mu^2+2\mu\theta-\theta^2}{2\tau^2}+\frac{-\mu^2+2\mu\bar{y}-\bar{y}^2}{2\sigma^2/n}]$$

From here we can simplify this equation:

$$\propto exp[\frac{-\mu^2+2\mu\theta-\theta^2}{2\tau^2}-\frac{\mu^2}{2\sigma^2/n}+\frac{2\mu\bar{y}-\bar{y}^2}{2\sigma^2/n}]$$

$$\propto exp[\frac{-\mu^2+2\mu\theta}{2\tau^2}-\frac{\theta^2}{2\tau^2}-\frac{\mu^2}{2\sigma^2/n}+\frac{2\mu\bar{y}}{2\sigma^2/n}-\frac{\bar{y}^2}{2\sigma^2/n}]$$

$$\propto exp[\frac{-\mu^2+2\mu\theta}{2\tau^2}]exp[\frac{-\mu^2+2\mu\bar{y}}{2\sigma^2/n}]$$

Now let us make sure that the there are common denominators so we can combine them into a single exponent:

$$\propto exp[\frac{(-\mu^2+2\mu\theta)\sigma^2/n}{2\tau^2\sigma^2/n}]exp[\frac{-\mu^2+2\mu\bar{y}-\bar{y}^2}{2\tau^2\sigma^2/n}]$$

$$\propto exp[\frac{(-\mu^2+2\mu\theta)\sigma^2+(-\mu^2+2\mu\bar{y})n\tau^2}{2\tau^2\sigma^2}]$$

Now we will combine like terms:

$$\propto exp[\frac{-\mu^2(n\tau^2+\sigma^2)+2\mu(\theta\sigma^2+\bar{y}n\tau^2)}{2\tau^2\sigma^2}]$$

$$\propto exp[\frac{-\mu^2+2\mu\frac{\theta\sigma^2+\bar{y}n\tau^2}{n\tau^2+\sigma^2}}{2(\tau^2\sigma^2)/(n\tau^2+\sigma^2)}]$$

Now we can bring back some constant which do not depend upon μ to complete the square in the numerator:

$$f(\mu|\vec{y}) \propto exp[\frac{-(\mu-\frac{\theta\sigma^2+\bar{y}n\tau^2}{n\tau^2+\sigma^2})^2}{2(\tau^2\sigma^2)/(n\tau^2+\sigma^2)}]$$

Then we complete the square, we get the kernel of the normal pdf for μ, $exp[-\frac{(\mu-v)^2}{2v^2}]$. By identifying the missing pieces denoted by v, we can conclude that the distribution for the posterior is:

$$\mu|\vec{y} \sim N(\frac{\theta\sigma^2 + \bar{y}n\tau^2}{n\tau^2 + \sigma^2}, \frac{\tau^2\sigma^2}{n\tau^2 + \sigma^2})$$

6 MCMC Simulation

Use RStan to simulate the posterior distribution for your example. Overlay a plot of the actual posterior based on the conjugate model to show how close the MCMC simulation is to the real thing. Use the MCMC simulated posterior to estimate the same clinically relevant posterior probability that you estimated above. Explain what is different about how you are finding this posterior probability as compared to the method you used based on the conjugate family analysis.

```r
data %>%
  select(BaselineWeight, EndWeight) %>%
  tbl_summary(type = everything() ~ "continuous2",
              statistic = list("EndWeight" ~ c(
      "{mean}, ({sd})",
      "{median} ({p25}, {p75})",
      "{min}, {max}"), "BaselineWeight" ~ c("{mean}, ({sd})"))
    ) %>%
  modify_caption("**Table 1. Data Summary**") %>%
  modify_footnote(everything() ~ "Weight in grams")
```

```
## Table printed with ‘knitr::kable()‘, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include ‘message = FALSE‘ in code chunk header.
```

Table 1: **Table 1. Data Summary**

Characteristic	**N = 45**
BaselineWeight	
Mean, (SD)	41.07, (1.14)
EndWeight	
Mean, (SD)	219, (72)
Median (IQR)	205 (167, 266)
Range	74, 373

As we can see from the table above, the baseline weight is comparable for all groups.

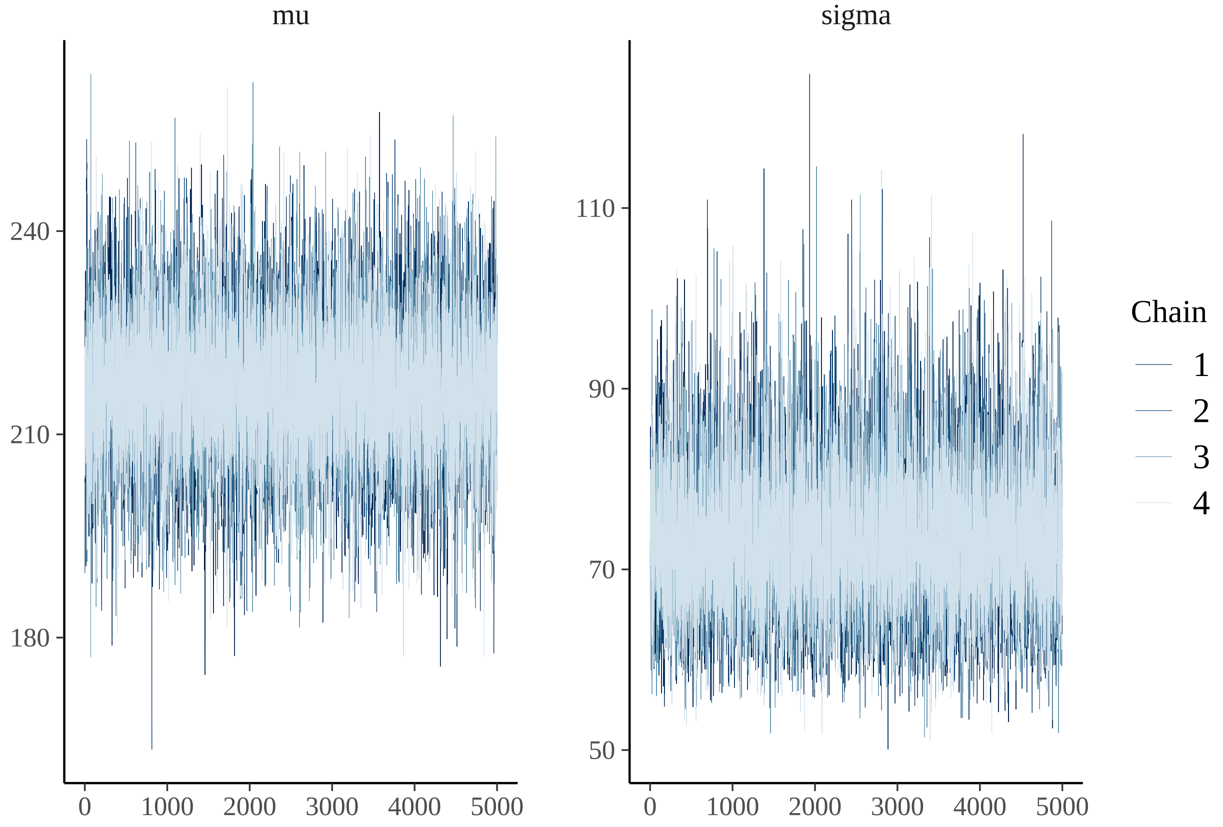
```r
library(rstan)
# Define the Stan Model
gp_model <- "
data {
  int<lower = 0> N;      // number of data points
  real Y[N];          // observed data
}
parameters {
  real mu;               // posterior mean
  real<lower=0> sigma;  // posterior standard deviation
}
model {
  mu ~ normal(190, 47.5);         // prior for mu
  Y ~ normal(mu, sigma);     // vectorized likelihood
}
"

# Simulate the posterior
sim <- stan(model_code = gp_model, data = list(N = length(data$EndWeight), Y = data$EndWeight), chains
```

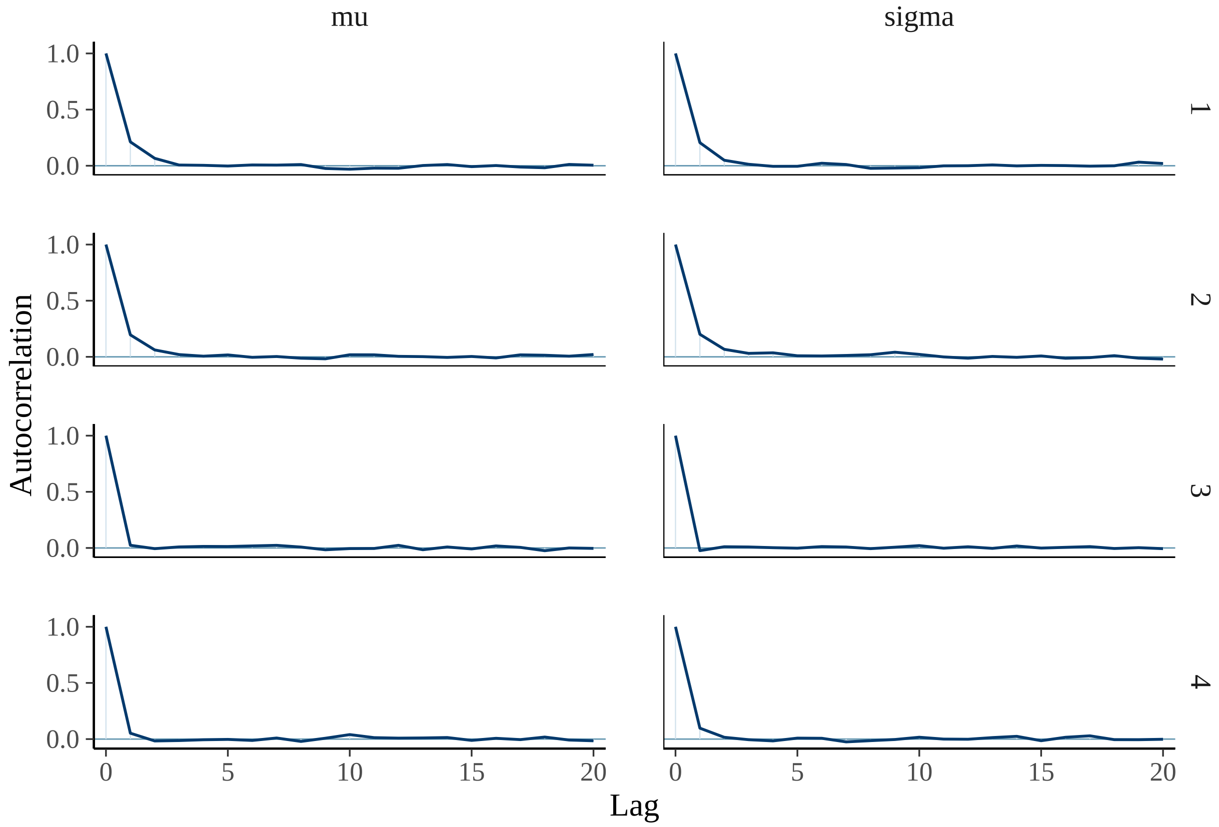
```
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 7.2e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.72 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 1: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 1: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 1: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 1: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 1: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 1: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 1: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 1: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 1: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 1: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 1: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.136 seconds (Warm-up)
## Chain 1:                0.109 seconds (Sampling)
## Chain 1:                0.245 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 7e-06 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.07 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 2: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 2: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 2: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 2: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 2: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 2: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 2: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 2: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 2: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 2: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 2: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.092 seconds (Warm-up)
## Chain 2:                0.137 seconds (Sampling)
## Chain 2:                0.229 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 6e-06 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.06 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 3: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 3: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 3: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 3: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 3: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 3: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 3: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 3: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 3: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 3: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 3: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.109 seconds (Warm-up)
## Chain 3:                0.086 seconds (Sampling)
## Chain 3:                0.195 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 6e-06 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.06 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 4: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 4: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 4: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 4: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 4: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 4: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 4: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 4: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 4: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 4: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 4: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 4:
```

```
## Chain 4:  Elapsed Time: 0.099 seconds (Warm-up)
## Chain 4:                0.089 seconds (Sampling)
## Chain 4:                0.188 seconds (Total)
## Chain 4:
```

```r
mcmc_trace(sim, pars = c("mu", "sigma"), size = 0.1)
```



```r
mcmc_acf(sim, pars = c("mu", "sigma"))
```



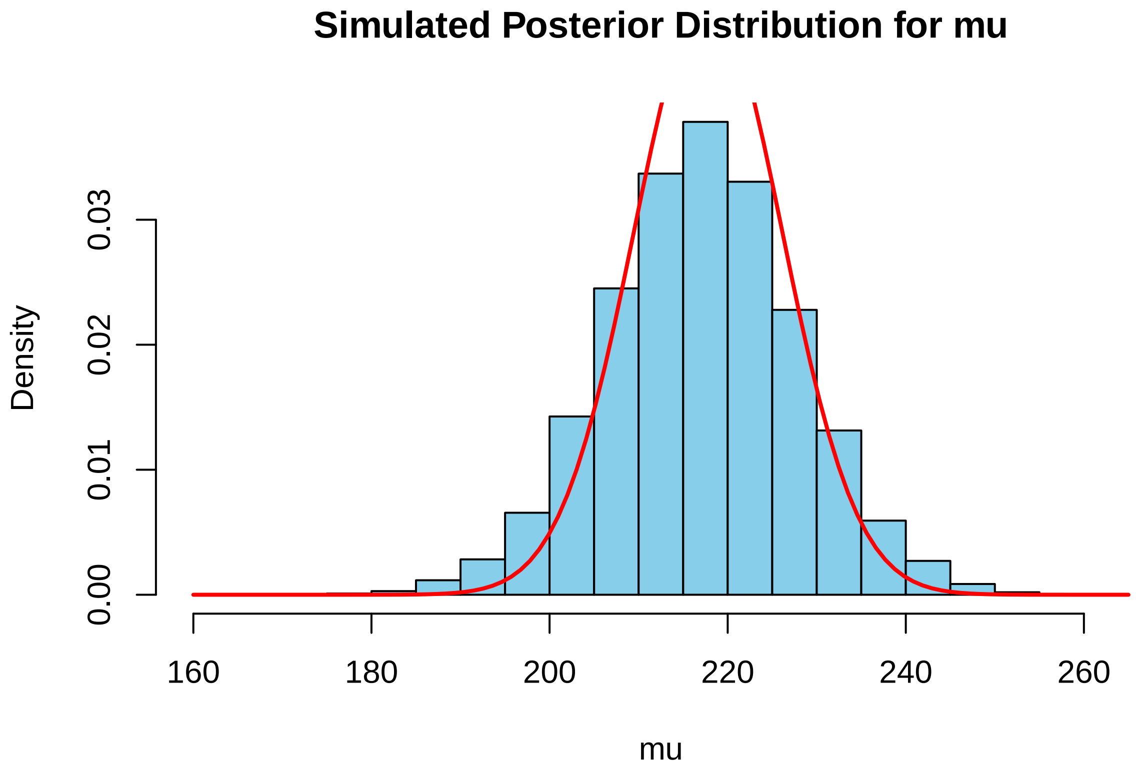
```
n_eff <- neff_ratio(sim, pars = c("mu", "sigma"))

# Calculate posterior parameters based on conjugate model
data_mean <- mean(data$EndWeight)
data_var <- var(data$EndWeight)
prior_mean <- 190
prior_var <- 47.5^2
likelihood_var <- 72

posterior_mean <- (prior_var * data_mean + likelihood_var * prior_mean) / (prior_var + likelihood_var)
posterior_var <- (prior_var * likelihood_var) / (prior_var + likelihood_var)

# Extract posterior parameters from MCMC simulation
posterior_samples <- extract(sim)
posterior_mu <- posterior_samples$mu
posterior_sigma <- posterior_samples$sigma

# Overlay histogram of simulated posterior with conjugate posterior
hist(posterior_mu, freq = FALSE, main = "Simulated Posterior Distribution for mu", xlab = "mu", col = "
curve(dnorm(x, mean = posterior_mean, sd = sqrt(posterior_var)), add = TRUE, col = "red", lwd = 2)
```



```
# Clinically relevant posterior probability estimation using simulated posterior
estimated_prob <- sum(posterior_mu > 209) / length(posterior_mu)
print(paste("Estimated probability of mu > 209 using MCMC simulated posterior:", round(estimated_prob, 
```

```
## [1] "Estimated probability of mu > 209 using MCMC simulated posterior: 0.78"
```

The effective sample size ratio of this simulation is 0.739, 0.733.

The difference between using the MCMC simulation and the conjugate family analysis is that the MCMC simulation uses repeated sampling to provide a numerical approximation of the posterior distribution, while the conjugate family analysis allows us to actually solve for the posterior distribution. MCMC simulations are especially useful when we are using complex models where solving mathematically for the posterior is not useful.

6.0.0.1 Contributions Sam: Background, Normal PDF, Normal Prior

Suz: Normal Likelihood, Normal-normal posterior

Dani: Simulation, data gathering and dataset set-up