Bayesian models and Markov chains

Mohar Sen

6/14/2020

Research topic: Sleep deprivation

Research Question
How does sleep deprivation impact reaction time?

The Study

- measure reaction time on Day 0
- restrict sleep to 3 hours per night
- measure reaction time on Day 3
- measure the change in reaction time

For subject i, let Y_i be the change in reaction time (in ms) after 3 sleep deprived nights. Of course, people react differently to sleep deprivation. It's reasonable to assume that Y_i are Normally distributed around some average m with standard deviation s

Y_i = change in reaction time(ms) for subject i

Assume
Y_i are Normally distributed around some average change in reaction time m with standard deviation s.

$Y_i \sim N(m, s^2)$

Prior model for parameter m
Y_i = change in reaction time (ms)
$Y_i \sim N(m, s^2)$
$m = averageY_i$

Prior information:

- with normal sleep, average reaction time is ~250 ms
- expect average to increase by ~50 m
- average is unlikely to decrease & unlikely to increase by more than ~150 ms

Thus, m ~ N(50, 25^2)

Also, * s>0 * with normal sleep, s.d. in reaction times is ~30 ms * s is equally likely to be anywhere from 0 to 200 ms

Thus, s ~ Unif(0,200)

Therefore, $Y_i \sim N(m, s^2)$ $m \sim N(50, 25^2)$ $s \sim Unif(0, 200)$

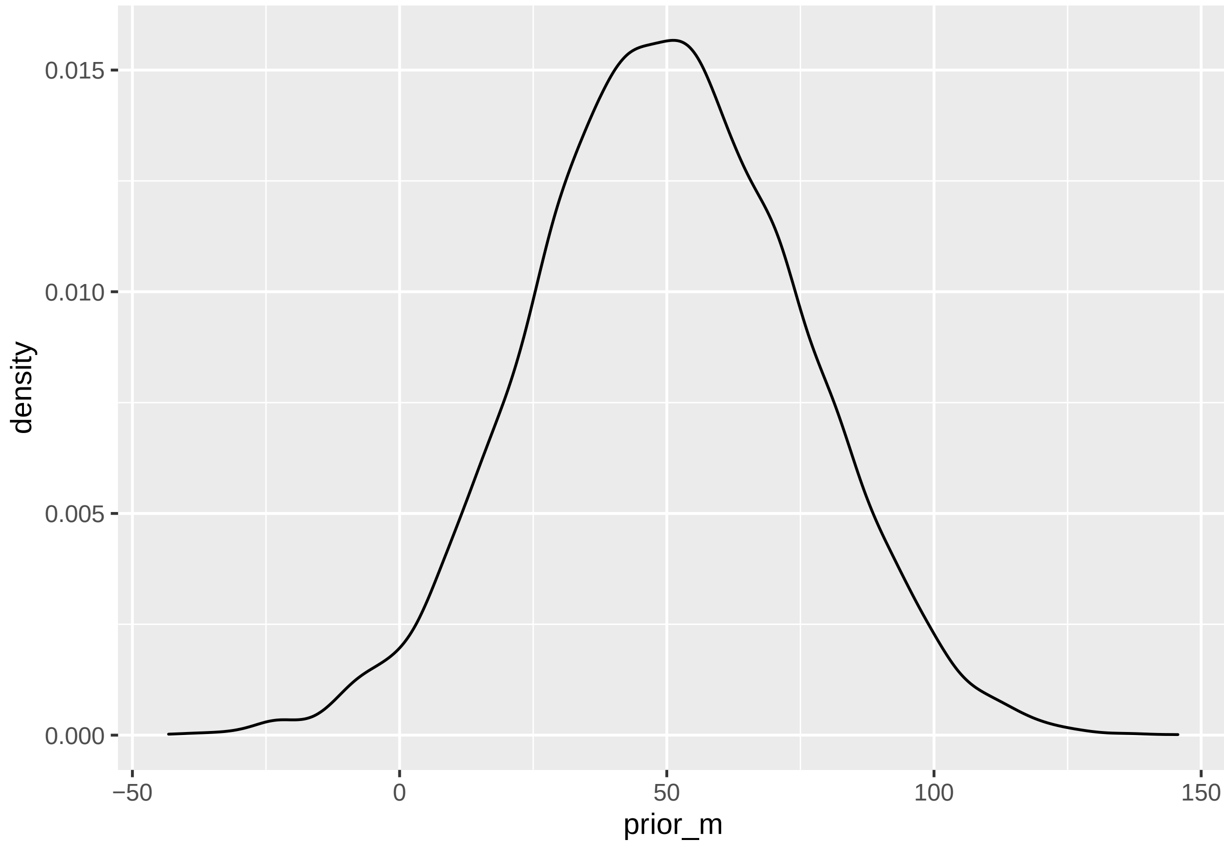
```r
library(ggplot2)
library(rjags)
library(tinytex)
options(tinytex.verbose=TRUE)
```

Normal-Normal priors

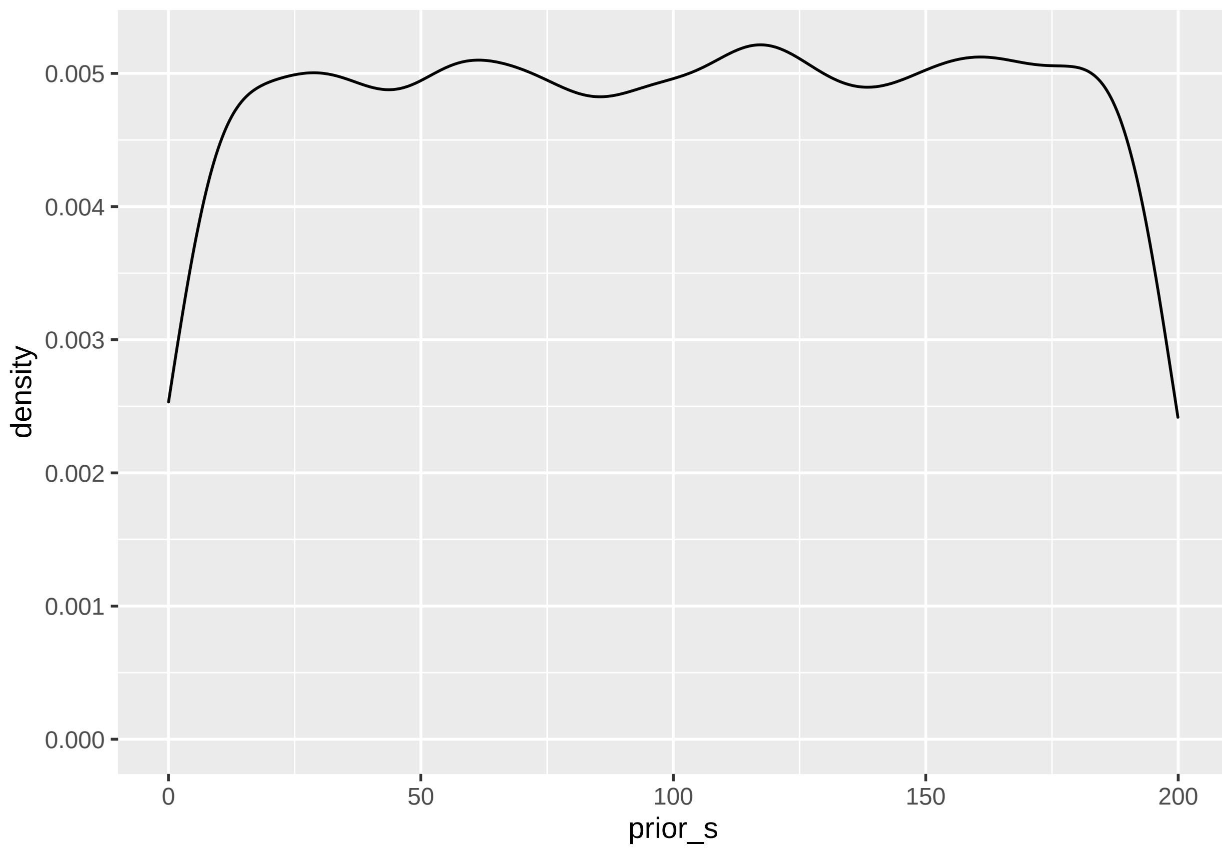
In the first step of your Bayesian analysis, you'll simulate the following prior models for parameters m and s:

- Use rnorm(n, mean, sd) to sample 10,000 draws from the m prior. Assign the output to prior_m.
- Use runif(n, min, max) to sample 10,000 draws from the s prior. Assign the output to prior_s.
- After storing these results in the samples data frame, construct a density plot of the prior_m samples and a density plot of the prior_s samples.

```r
# Take 10000 samples from the m prior
prior_m <- rnorm(n=10000, mean=50, sd=25)

# Take 10000 samples from the s prior
prior_s <- runif(n=10000, min=0, max=200)

# Store samples in a data frame
samples <- data.frame(prior_m, prior_s)

# Density plots of the prior_m & prior_s samples
ggplot(samples, aes(x = prior_m)) +
    geom_density()
```




```r
ggplot(samples, aes(x = prior_s)) +
    geom_density()
```



The distributions of these random samples approximate the features of your Normal prior for m and Uniform prior for s.

Sleep study data

Researchers enrolled 18 subjects in a sleep deprivation study. Their observed `sleep_study` data are loaded in the workspace. These data contain the `day_0` reaction times and day_3 reaction times after 3 sleep deprived nights for each subject.

You will define and explore `diff_3`, the observed difference in reaction times for each subject. This will require the `mutate()` & `summarize()` functions. For example, the following would add variable `day_0_s`, `day_0` reaction times in seconds, to sleep_study:

```
sleep_study <- sleep_study %>%
    mutate(day_0_s = day_0 * 0.001)
```

You can then summarize() the day_0_s values, here by their minimum & maximum:

```
sleep_study  %>%
    summarize(min(day_0_s), max(day_0_s))
```

```
sleep_study <- readr::read_csv('data/sleep_study.csv')
library(dplyr)
```

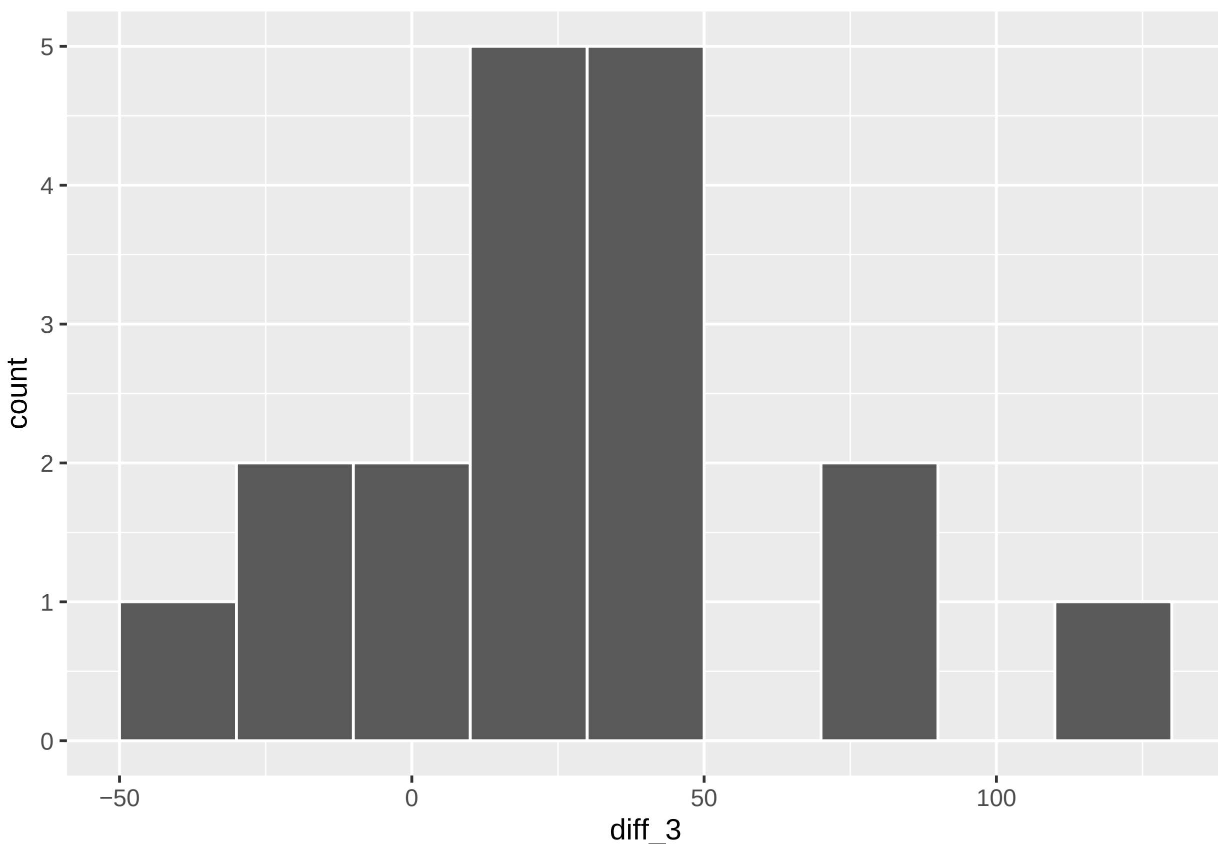
Instructions

- Check out the first 6 rows of sleep_study.
- Define a new sleep_study variable diff_3, the day_3 minus the day_0 reaction times.
- Use `ggplot()` with a geom_histogram() layer to construct a histogram of the diff_3 data.
- `summarize()` the mean and standard deviation of the diff_3 observations.

```
# Check out the first 6 rows of sleep_study
head(sleep_study)
```

```
## # A tibble: 6 x 3
##   subject day_0 day_3
##     <dbl> <dbl> <dbl>
## 1     308  250.  321.
## 2     309  223.  205.
## 3     310  199.  233.
## 4     330  322.  285.
## 5     331  288.  320.
## 6     332  235.  310.
```

```
# Define diff_3
sleep_study <- sleep_study %>%
    mutate(diff_3 = day_3 - day_0)

# Histogram of diff_3
ggplot(sleep_study, aes(x = diff_3)) +
    geom_histogram(binwidth = 20, color = "white")
```



```r
# Mean and standard deviation of diff_3
sleep_study %>%
    summarize(mean(diff_3), sd(diff_3))
```

```
## # A tibble: 1 x 2
##   `mean(diff_3)` `sd(diff_3)`
##            <dbl>        <dbl>
## 1           26.3         37.2
```

Reaction times increased by an average of ~26 ms with a standard deviation of ~37 ms. Further, only 4 of the 18 test subjects had faster reaction times on day 3 than on day 0. Though not in perfect agreement about the degree to which the average reaction time changes under sleep deprivation, both the likelihood and prior are consistent with the hypothesis that the average increases relative to reaction time under normal sleep conditions.

Define, compile, & simulate the Normal-Normal

Upon observing the change in reaction time Y_i for each of the 18 subjects i enrolled in the sleep study, you can update your posterior model of the effect of sleep deprivation on reaction time. This requires the combination of insight from the likelihood and prior models:

- likelihood: $Y_i \sim N(m, s^2)$
- priors: $m \sim N(50, 25^2)$ and $s \sim Unif(0, 200)$

In this series of exercises, you'll **define**, **compile**, and **simulate** your Bayesian posterior.

Step 1: Define DEFINE your Bayesian model and store the model string as `sleep_model`. In doing so, note that:

- `dnorm(a, b)` defines a $N(a, b-1)$ model with precision (ie. inverse variance) b.
- `dunif(a,b)` defines a $Unif(a, b)$ model.
- The model of Y_i depends upon m and s. The number of subjects i is defined by `length(Y)`.

```r
# DEFINE the model
sleep_model <- "model{
    # Likelihood model for Y[i]
    for(i in 1:length(Y)) {
        Y[i] ~ dnorm(m, s^(-2))
    }

    # Prior models for m and s
    m ~ dnorm(50, 25^(-2))
    s ~ dunif(0, 200)
}"
```

Step 2: Compile COMPILE sleep_model using jags.model():

- Establish a `textConnection()` to `sleep_model` and provide the observed vector of `Y[i]` data from `sleep_study`. (Ignore `inits` for now!)
- Store the output in a jags object named `sleep_jags`.

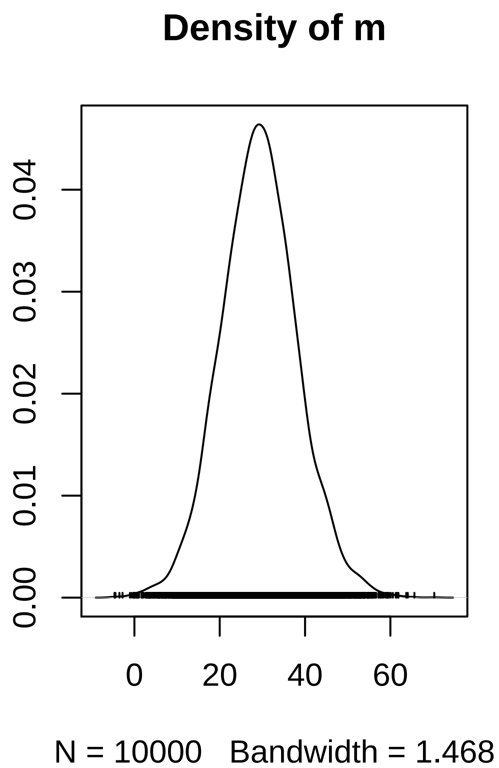
```r
# COMPILE the model
sleep_jags <- jags.model(
  textConnection(sleep_model),
  data = list(Y = sleep_study$diff_3),
  inits = list(.RNG.name = "base::Wichmann-Hill", .RNG.seed = 1989)
)
```

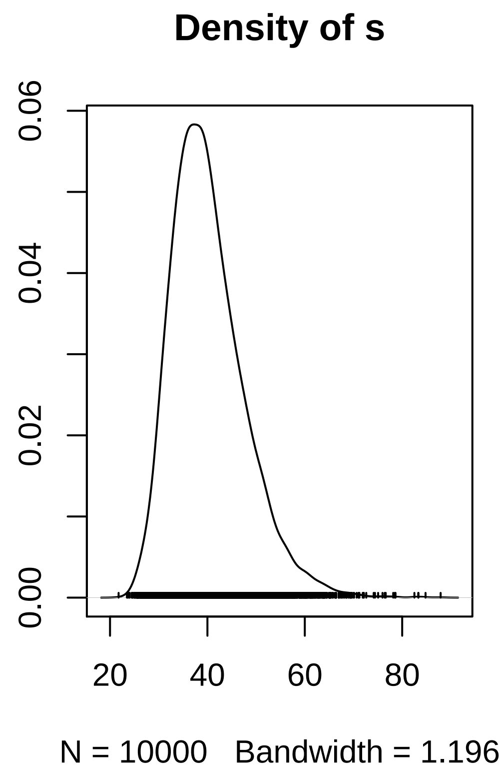
```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 18
##    Unobserved stochastic nodes: 2
##    Total graph size: 28
##
## Initializing model
```

Step 3: Simulate SIMULATE a sample of 10,000 draws from the posterior model of m and s

- The required `coda.samples()` function takes 3 arguments: the compiled model, variable.names (the model parameter(s)), `n.iter` (sample size). Store this `mcmc.list` in `sleep_sim`.
- Construct a density `plot()` of the posterior samples in `sleep_sim`.

```r
# SIMULATE the posterior
sleep_sim <- coda.samples(model = sleep_jags, variable.names = c("m", "s"), n.iter = 10000)

# PLOT the posterior
plot(sleep_sim, trace = FALSE)
```





Nice work!

Your posterior model is more narrow and lies almost entirely above 0, thus you're more confident that the average reaction time increases under sleep deprivation. Further, the location of the posterior is below that of the prior. This reflects the strong insight from the observed sleep study data in which the increase in average reaction time was only ~26 ms.

Markov chains

The sample of m values in `sleep_sim` is a dependent Markov chain, the distribution of which converges to the posterior. You will examine the contents of `sleep_sim` and, to have finer control over your analysis, store the contents in a data frame.

```r
# Check out the head of sleep_sim
head(sleep_sim)
```

```
## [[1]]
## Markov Chain Monte Carlo (MCMC) output:
## Start = 1001
## End = 1007
## Thinning interval = 1
##              m        s
## [1,] 17.25796 31.46256
## [2,] 34.58469 37.88655
## [3,] 36.45480 39.58056
```

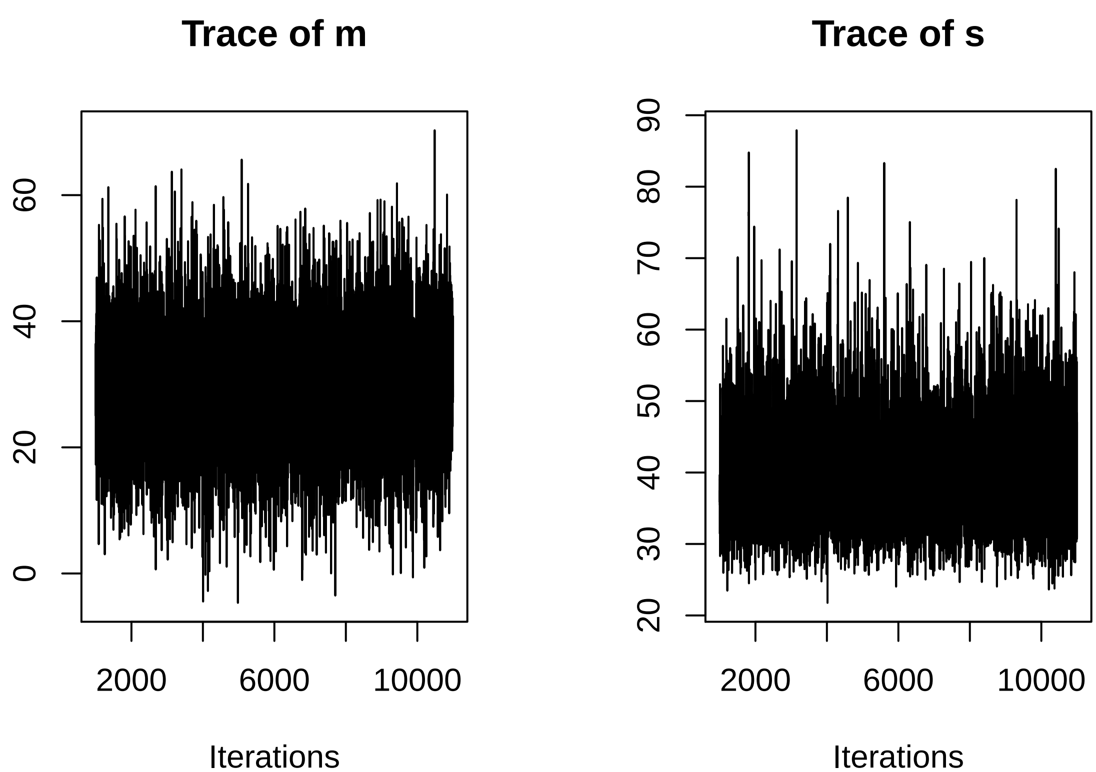
```
## [4,] 25.00971 39.69494
## [5,] 29.95475 35.90001
## [6,] 28.43894 37.46466
## [7,] 38.32427 35.44081
##
## attr(,"class")
## [1] "mcmc.list"
```

```
# Store the chains in a data frame
sleep_chains <- data.frame(as.matrix(sleep_sim), iter=1:10000)

# Check out the head of sleep_chains
head(sleep_chains)
```

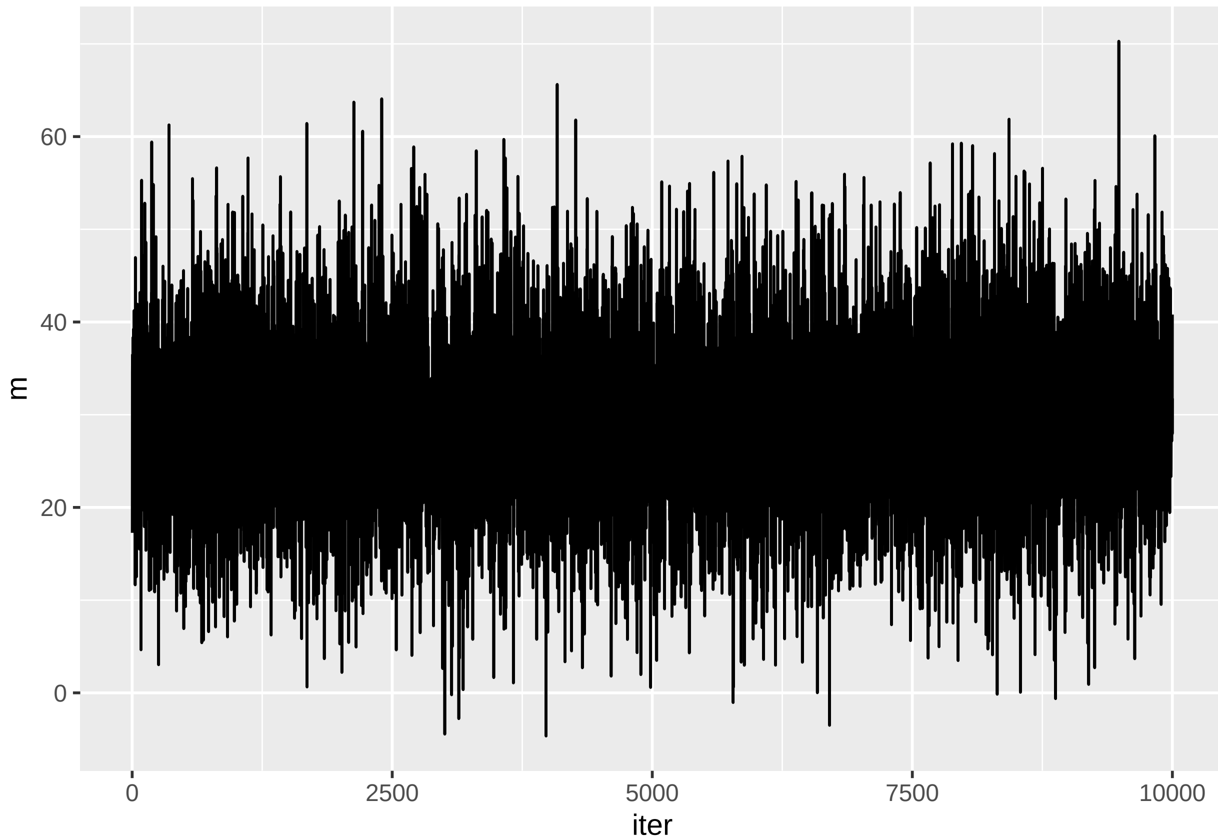
```
##          m        s iter
## 1 17.25796 31.46256    1
## 2 34.58469 37.88655    2
## 3 36.45480 39.58056    3
## 4 25.00971 39.69494    4
## 5 29.95475 35.90001    5
## 6 28.43894 37.46466    6
```

Next, you'll visualize the contents of these Markov chains.

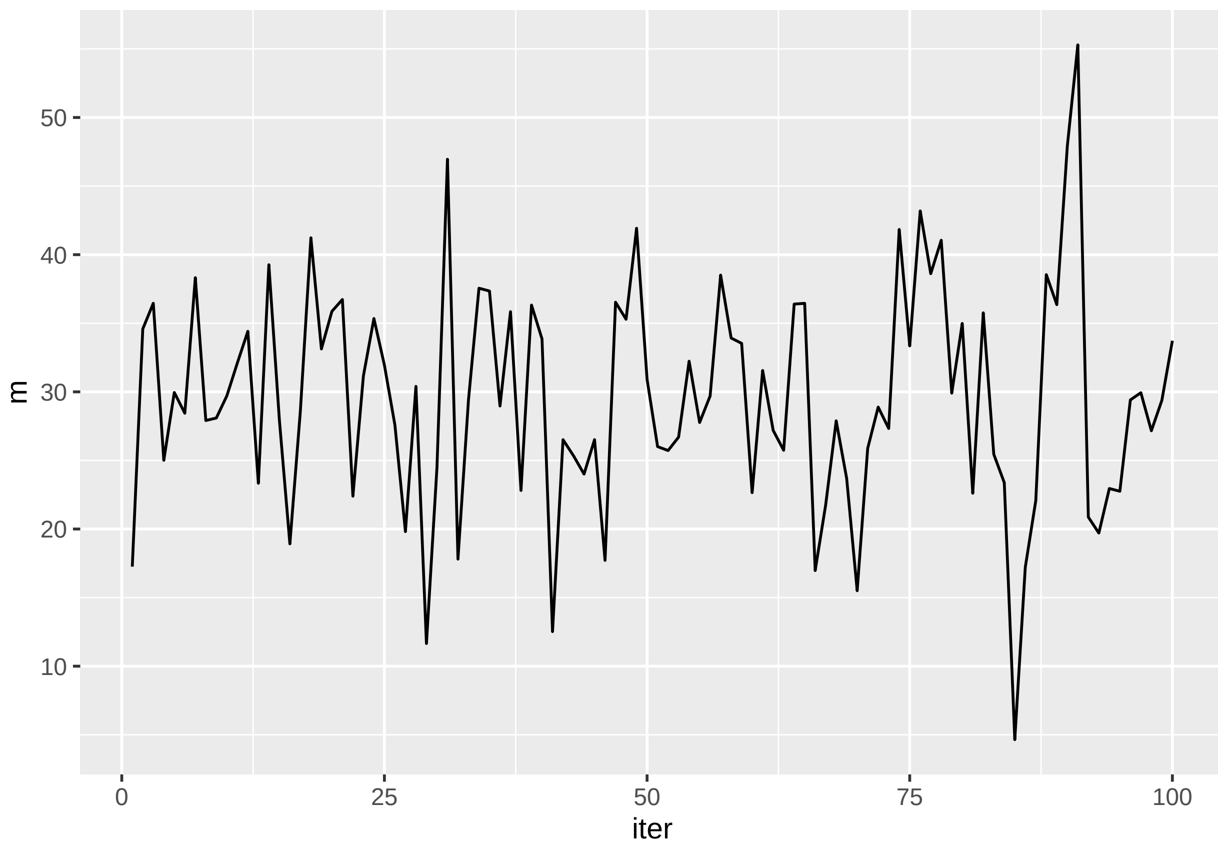
```
# Use plot() to construct trace plots of the m and s chains
plot(sleep_sim, density = FALSE)
```



```r
# Use ggplot() to construct a trace plot of the m chain
ggplot(sleep_chains, aes(x = iter, y = m)) +
    geom_line()
```




```r
# Trace plot the first 100 iterations of the m chain
ggplot(sleep_chains[sleep_chains$iter<101,], aes(x = iter, y = m)) +
    geom_line()
```



Note that the longitudinal behavior of the chain appears quite random and that the trend remains relatively constant. This is a good thing. It indicates that the Markov chain (likely) converges quickly to the posterior distribution of m.

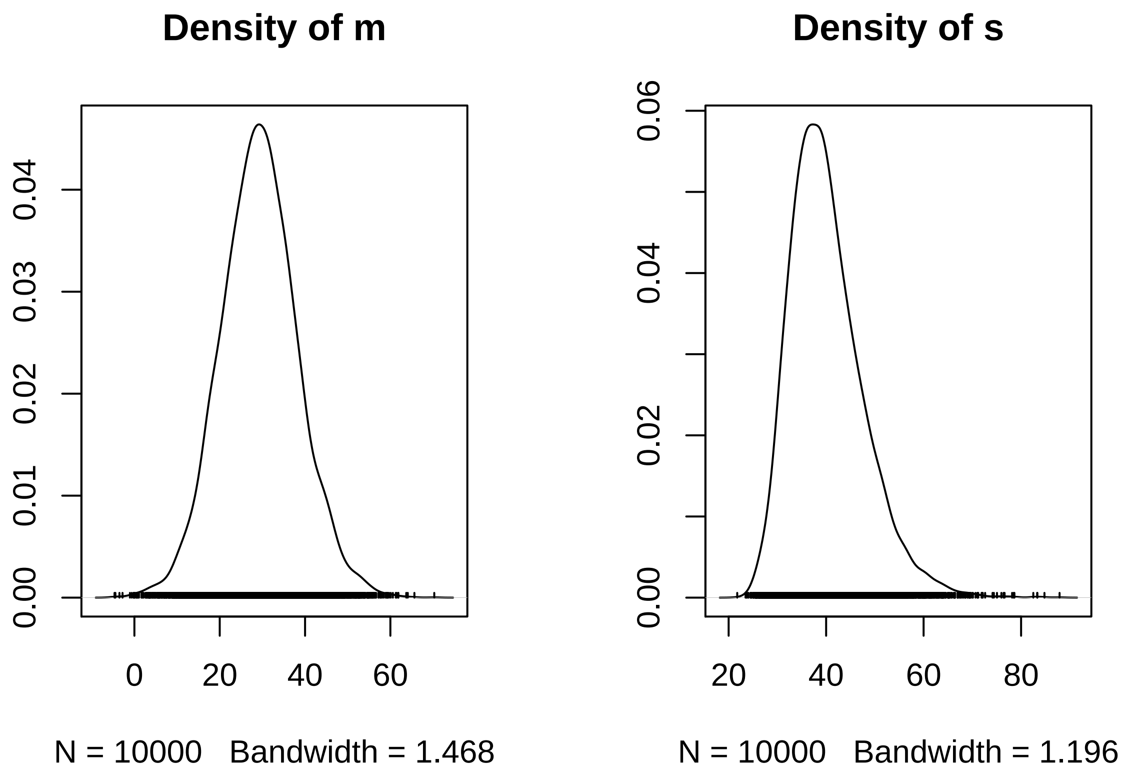
Markov chain density plots

Whereas a trace plot captures a Markov chain's longitudinal behavior, a density plot illustrates the final distribution of the chain values. In turn, the density plot provides an approximation of the posterior model. You will construct and examine density plots of the m Markov chain below.

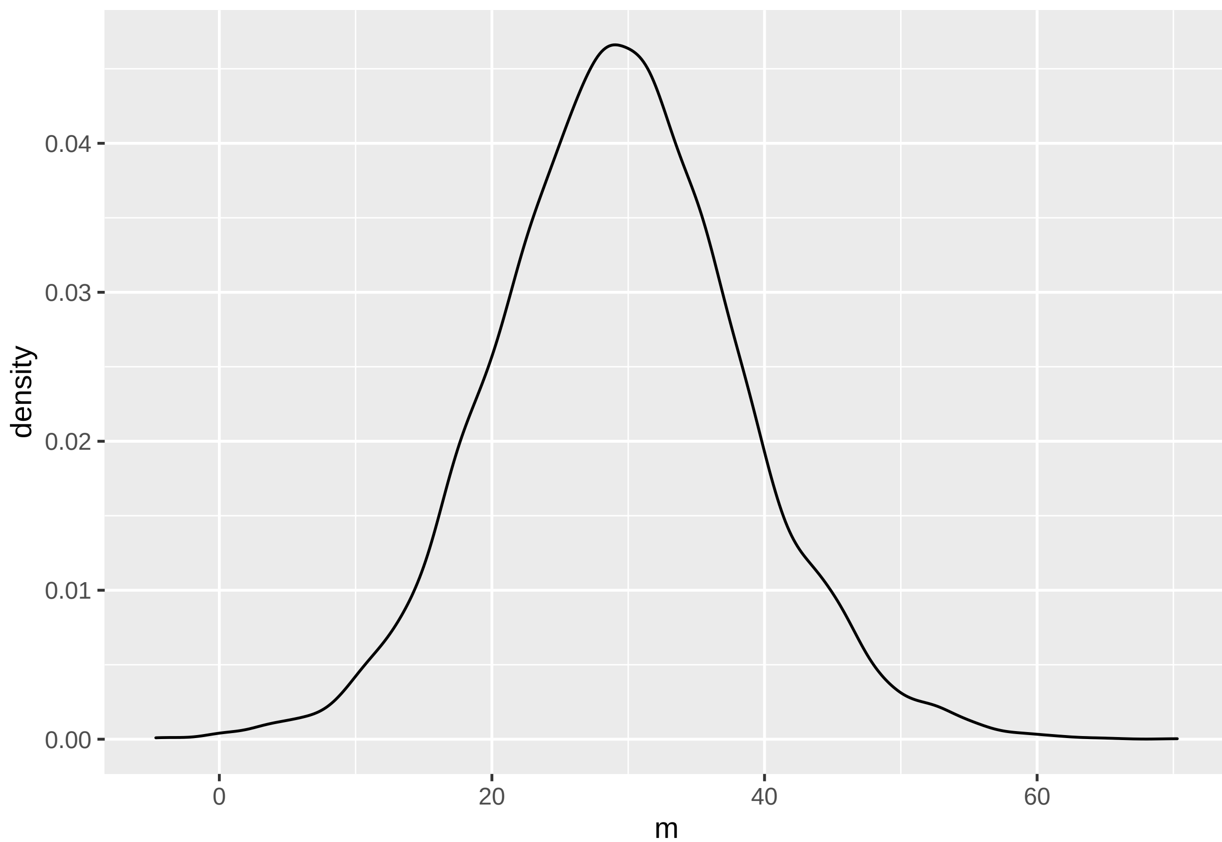
Instructions

- Apply plot() to sleep_sim with trace = FALSE to construct density plots for the m and s chains.
- Apply ggplot() to sleep_chains to re-construct a density plot of the m chain.

```
# Use plot() to construct density plots of the m and s chains
plot(sleep_sim, trace = FALSE)
```




```r
# Use ggplot() to construct a density plot of the m chain
ggplot(sleep_chains, aes(x = m)) +
    geom_density()
```



Remember, these plots *approximate* the posterior models of m and s