Bayesian Inference and Prediction

Mohar Sen

6/14/2020

Chapter 3 goals

- Engineer a simple Bayesian regression model
- Define, compile, and simulate regression models in RJAGS
- Use Markov chain simulation output for posterior inference & prediction

```
library(ggplot2)
library(rjags)
```

```
## Loading required package: coda
```

```
## Linked to JAGS 4.3.0
```

```
## Loaded modules: basemod,bugs
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Regression priors

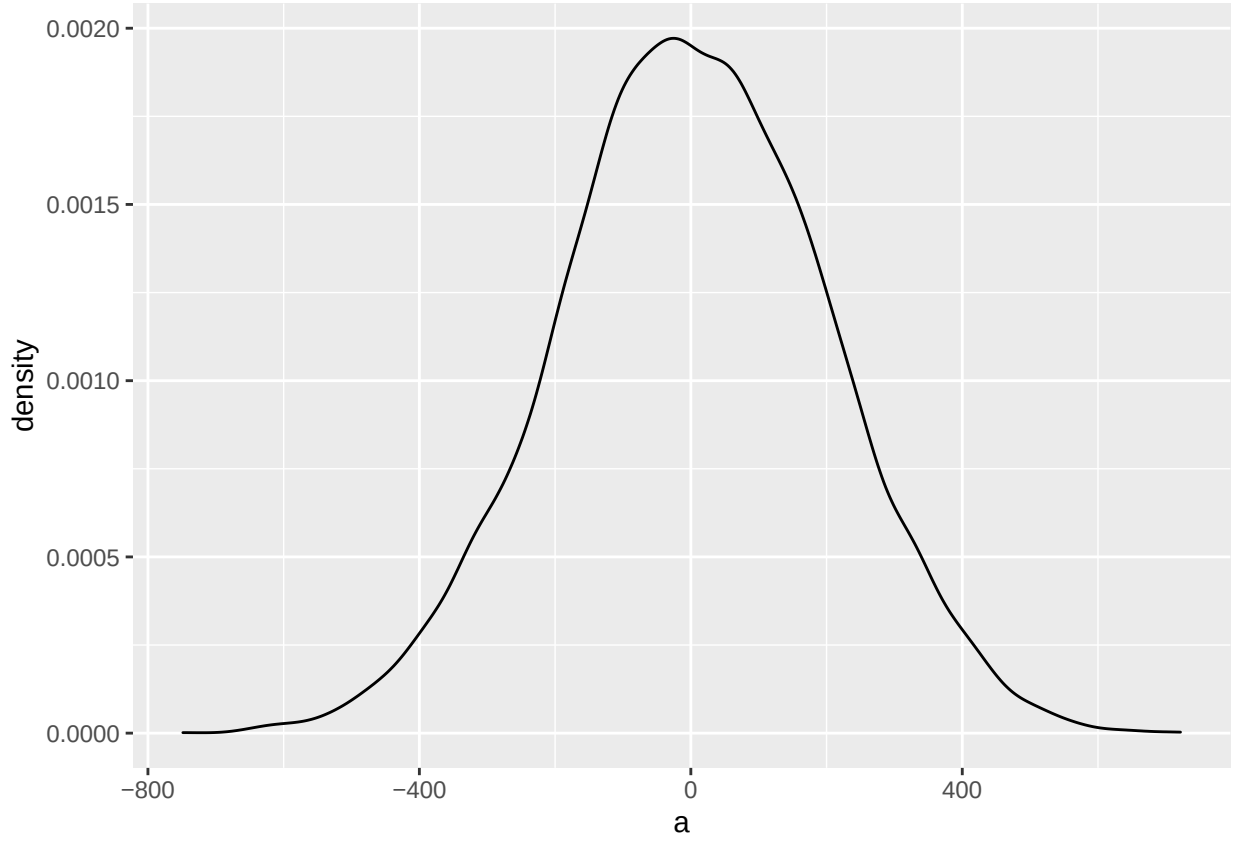
Let Y_i be the weight (in kg) of subject i. Past studies have shown that weight is linearly related to height X_i (in cm). The average weight m_i among adults of any shared height X_i can be written as $m_i = a + bX_i$. But height isn't a perfect predictor of weight - individuals vary from the trend. To this end, it's reasonable to assume that Yi are Normally distributed around m_i with residual standard deviation s: $Y_i \sim N(m_i, s^2)$

Note the 3 parameters in the model of weight by height: intercept a, slope b, & standard deviation s. In the first step of your Bayesian analysis, you will simulate the following prior models for these parameters: $a \sim N(0, 200^2)$, $b \sim N(1, 0.5^2)$, and $c \sim Unif(0, 20)$.

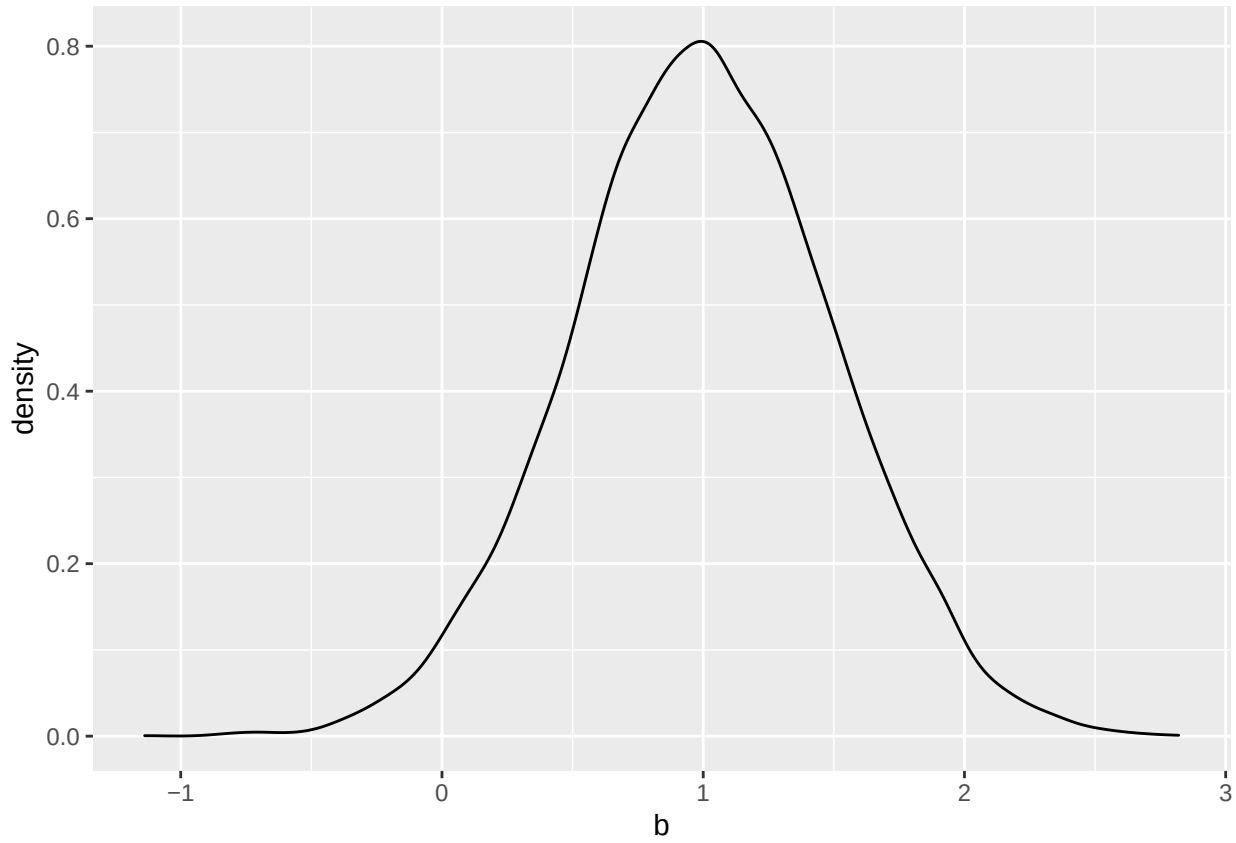
Instructions

- Sample 10,000 draws from each of the a, b, and s priors. Assign the output to a, b, and s. These are subsequently combined in the samples data frame along with set = 1:10000, an indicator of the draw numbers.
- Construct separate density plots of each of the a, b, and s samples.

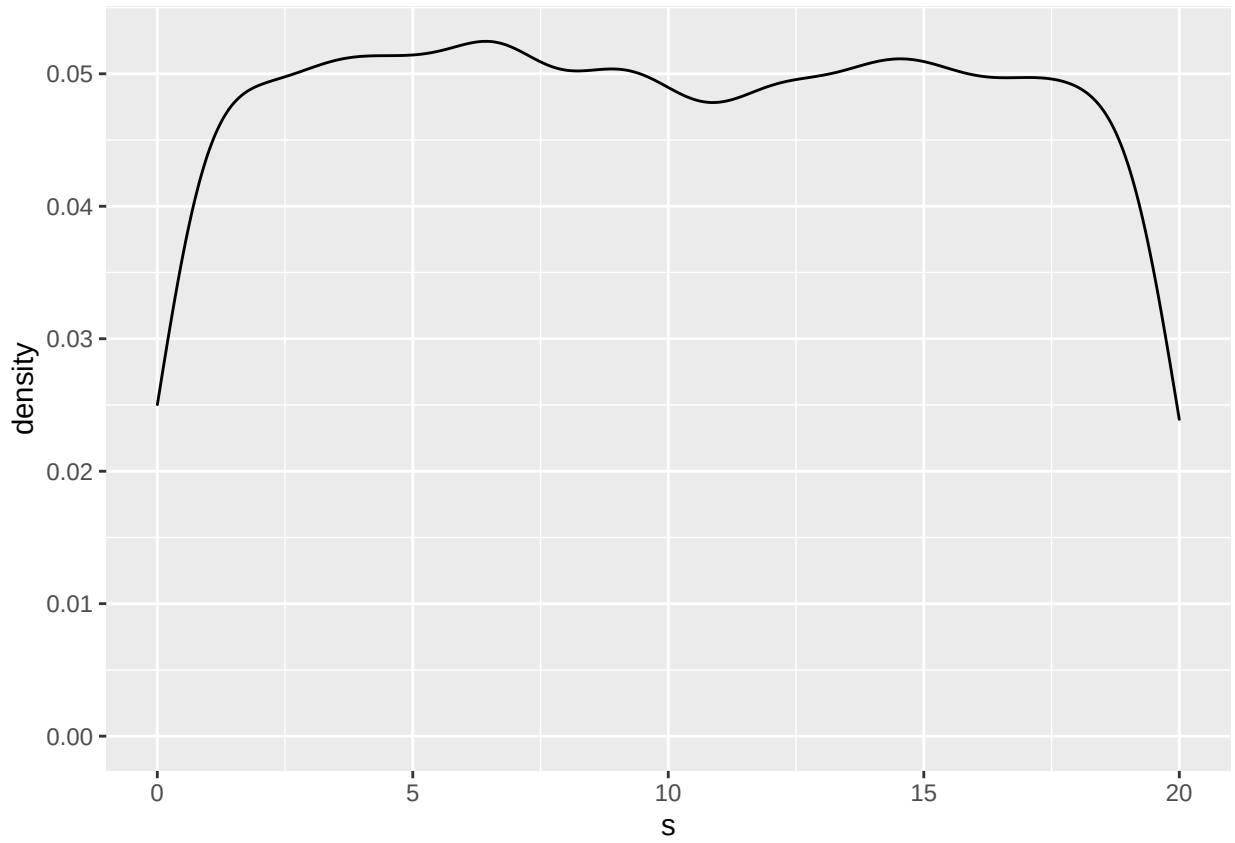
```
# Take 10000 samples from the a, b, & s priors
a <- rnorm(10000,0,200)
b <- rnorm(10000,1,0.5)
s <- runif(10000,0,20)

# Store samples in a data frame
samples <- data.frame(set = 1:10000, a, b, s)

# Construct density plots of the prior samples
ggplot(samples, aes(x = a)) +
    geom_density()
```




```
ggplot(samples, aes(x = b)) +
    geom_density()
```




```
ggplot(samples, aes(x = s)) +
    geom_density()
```



These simulations approximate your prior models of each separate model parameter. There's likely a positive association between weight & height ($b > 0$) but more uncertainty about the intercept a. Further, at any given height, the typical deviation of individuals' weights from the trend is equally likely to be anywhere between 0 and 20 kg.

Visualizing the regression priors

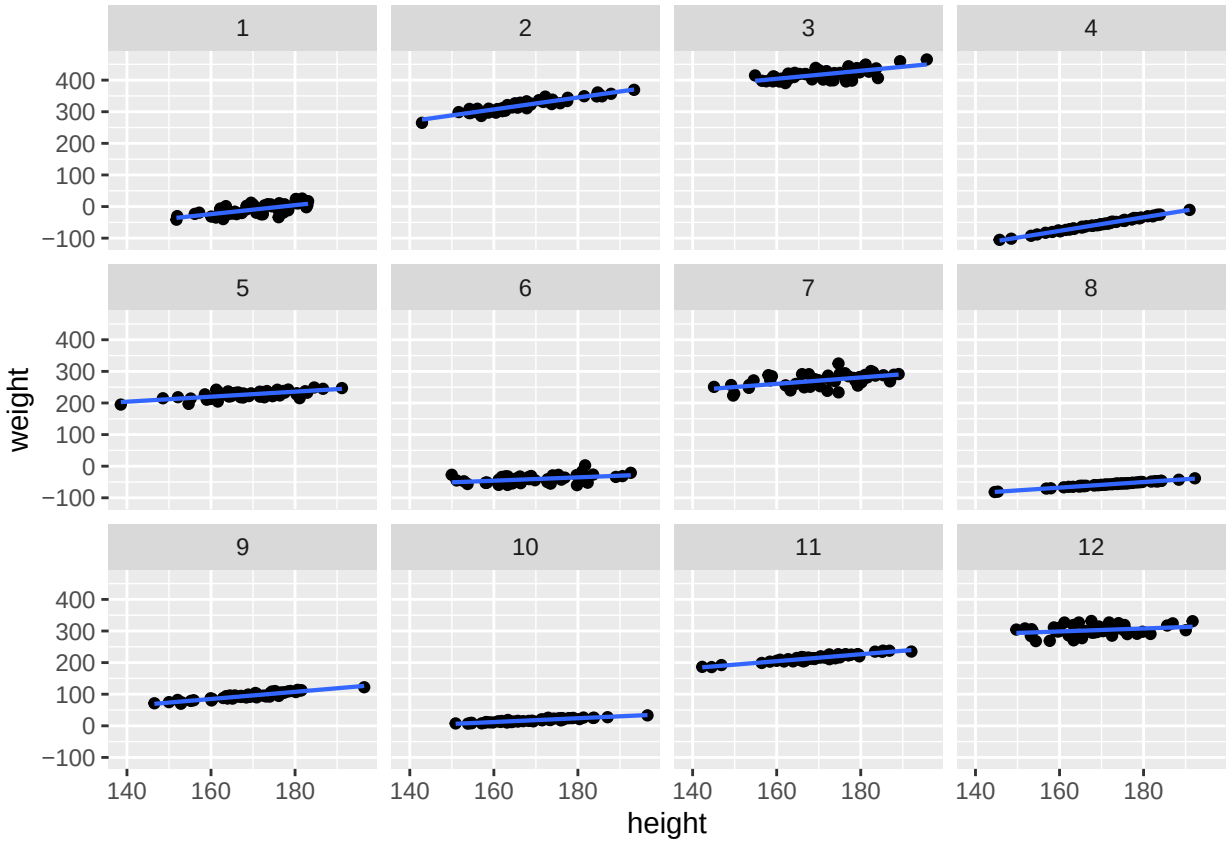
In the previous exercise, you simulated 10,000 samples for each parameter (a, b, s) in the Bayesian regression model of weight Y by height X: $Y_i \sim N(m_i, s^2)$ with mean $m = a + bX$. The set of a, b, and s values in each row of samples represents a prior plausible regression scenario. To explore the scope of these prior scenarios, you will simulate 50 pairs of height and weight values from each of the first 12 sets of prior parameters a, b, and s.

Instructions

- Create a data frame prior_simulation which includes n = 50 replicates of the first 12 sets of prior parameters in samples (600 rows in total!).
- For each of the 600 prior_simulation rows:
  - Simulate a height value from a $N(170, 10^2)$ model.
  - Simulate a weight value from $N(a + bX, s^2)$
  - where X is height and (a, b, s) are the prior parameter set.
- You now have 50 simulated height and weight pairs for each of the 12 parameter sets. Use `ggplot()` to construct a scatterplot of these 50 pairs for each set of parameter values. Be sure to put weight on the y-axis!

```
# Replicate the first 12 parameter sets 50 times each
prior_scenarios_rep <- bind_rows(replicate(n = 50, expr = samples[1:12, ], simplify = FALSE))

# Simulate 50 height & weight data points for each parameter set
prior_simulation <- prior_scenarios_rep %>%
    mutate(height = rnorm(n = 600, mean = 170, sd = 10)) %>%
    mutate(weight = rnorm(n = 600, mean = a+b*height, sd = s))

# Plot the simulated data & regression model for each parameter set
ggplot(prior_simulation, aes(x = height, y = weight)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE, size = 0.75) +
    facet_wrap(~ set)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



These 12 plots demonstrate the range of prior plausible models. These models have different intercepts, slopes, and residual standard deviations. Almost all of the models have positive slopes, demonstrating the prior information that there is likely a positive association between weight & height. Given your vague prior for a, some of these models are even biologically impossible!

Weight & height data

The `bdims` data set from the `openintro` package contains physical measurements on a sample of 507 individuals, including their weight in kg (wgt) and height in cm (hgt). You will use these data to build insights

into the relationship between weight and height.

```
library(openintro)
```

```
## Please visit openintro.org for free statistics materials
```

```
##
## Attaching package: 'openintro'
```

```
## The following object is masked from 'package:ggplot2':
##
##     diamonds
```

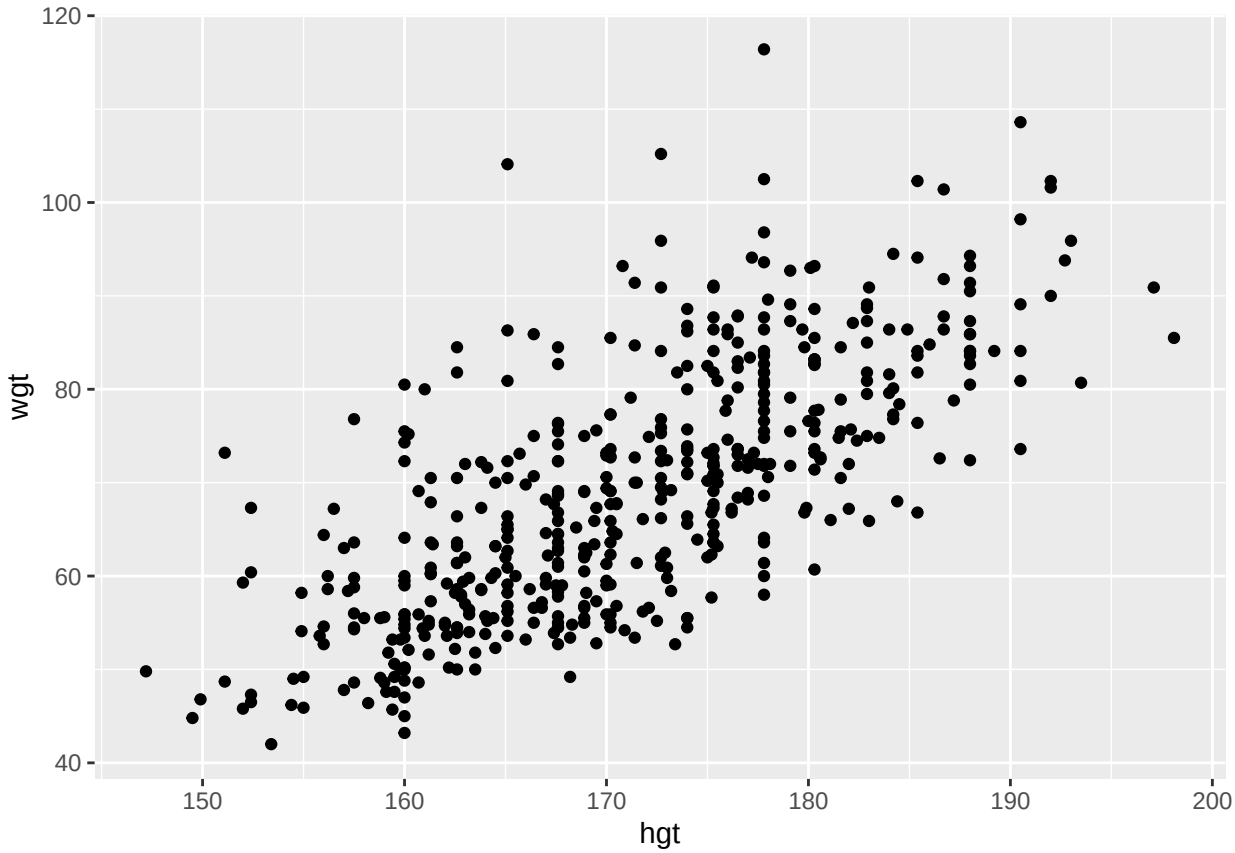
```
## The following objects are masked from 'package:datasets':
##
##     cars, trees
```

```
data("bdims")
```

instructions

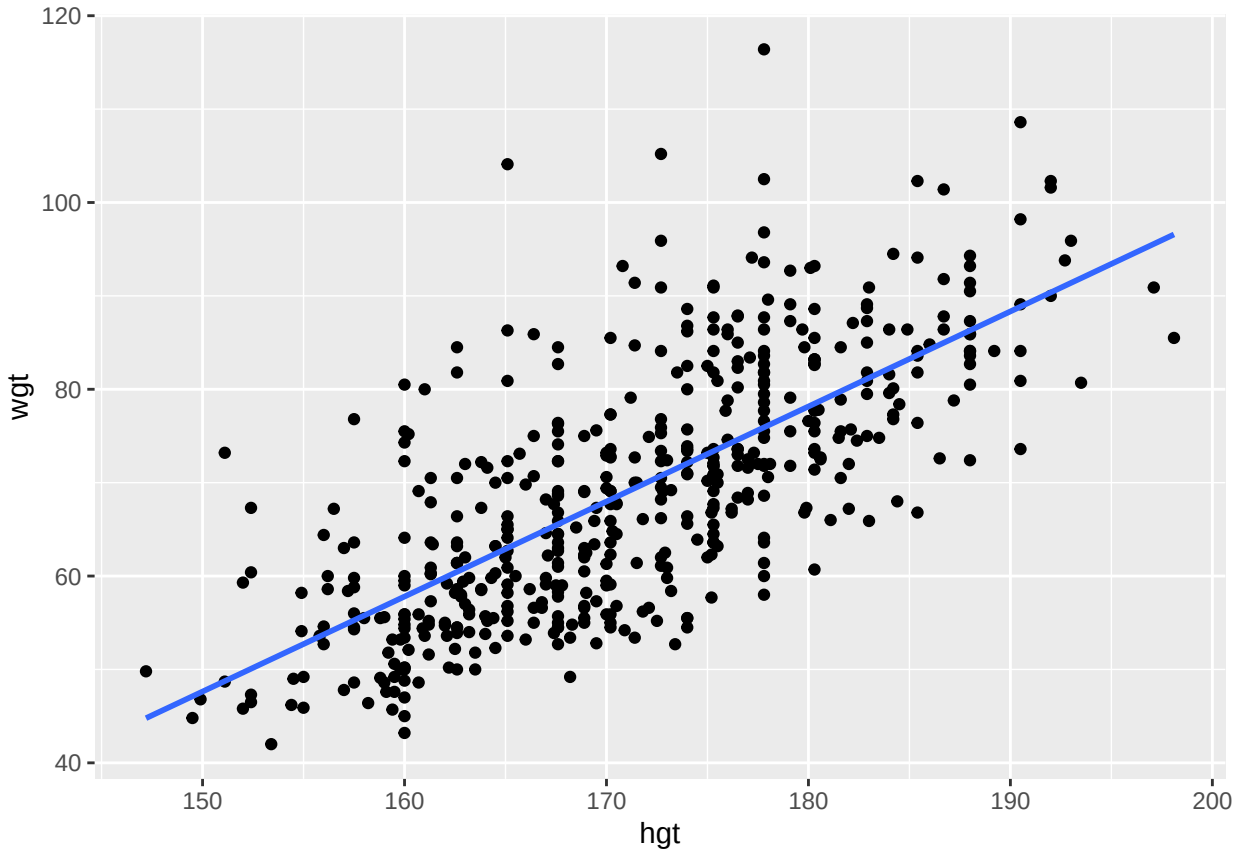
- Construct a scatterplot of `wgt` (y-axis) vs `hgt` (x-axis) using `ggplot()` with a `geom_point()` layer.
- Construct a scatterplot of wgt vs hgt which includes a `geom_smooth()` of the linear relationship between these 2 variables.

```
# Construct a scatterplot of wgt vs hgt
ggplot(bdims, aes(x = hgt, y = wgt)) +
    geom_point()
```




```
# Add a model smooth
ggplot(bdims, aes(x = hgt, y = wgt)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



These data support your prior information about a positive association between weight and height. With insights from the priors and data in place, you're ready to simulate the posterior regression model in RJAGS!

Define, compile, & simulate the regression model

Upon observing the relationship between weight Y_i and height X_i for the 507 subjects i in the bdims data set, you can update your posterior model of this relationship. To build your posterior, you must combine your insights from the likelihood and priors:

- likelihood: $Y_i \sim N(m_i, s^2)$ where $m_i = a + bX_i$
- priors: $a \sim N(0, 200^2)$, $b \sim N(1, 0.5^2)$ and $s \sim Unif(0, 20)$

Instructions 1/3

DEFINE your Bayesian model.

- Define the likelihood model of `Y[i]` given `m[i]` and `s` where `m[i] <- a + b * X[i]`.
- Specify the priors for `a`, `b`, and `s`.
- Store the model string as `weight_model`.

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

```
        m[i] <- a + b*X[i]
    }

    # Prior models for m and s
    a ~ dnorm(0, 200^(-2))
    b ~ dnorm(1, 0.5^(-2))
    s ~ dunif(0, 20)
}"
```

Instructions 2/3

COMPILE `weight_model` using `jags.model()`:

- Establish a `textConnection()` to `weight_model`.
- Provide the observed vectors of `Y` and `X` data from `bdims`.
- Store the output in a jags object named `weight_jags`.

```
# COMPILE the model
weight_jags <- jags.model(
  textConnection(weight_model),
  data = list(Y = bdims$wgt, X = bdims$hgt),
  inits = list(.RNG.name = "base::Wichmann-Hill", .RNG.seed = 1989)
)
```

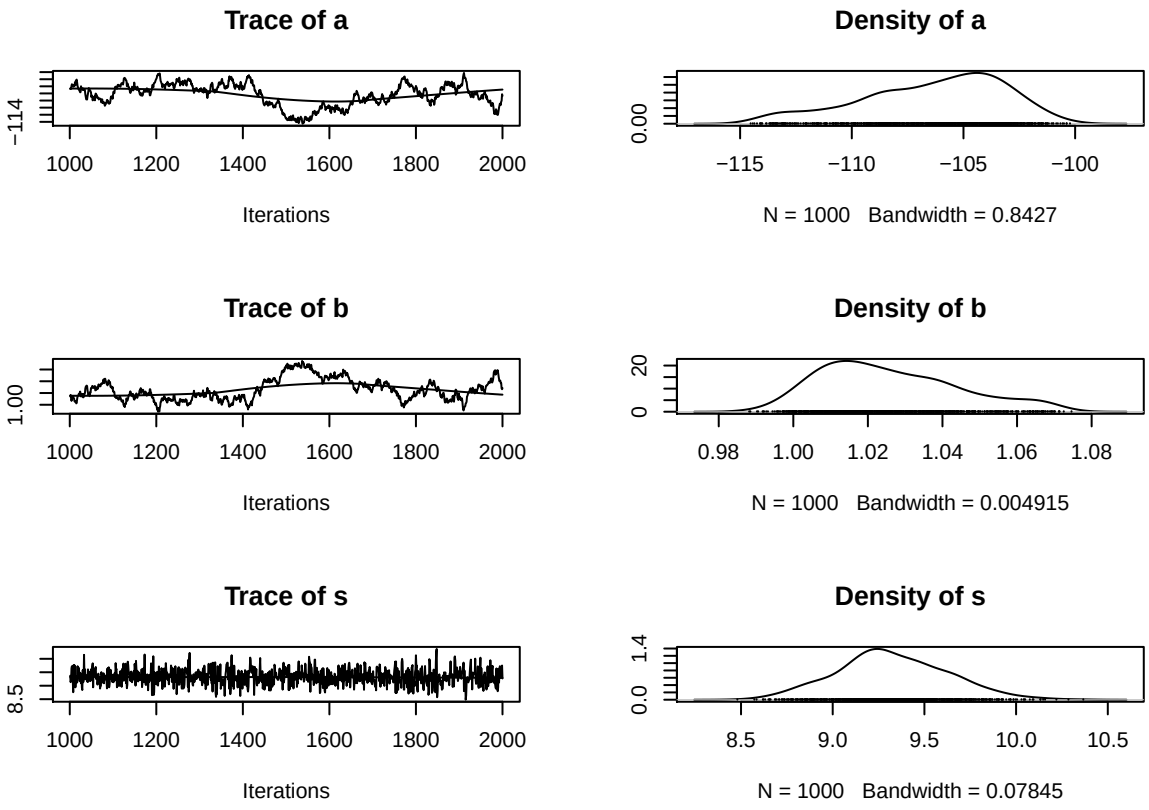
```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 507
##    Unobserved stochastic nodes: 3
##    Total graph size: 1321
##
## Initializing model
```

Instructions 3/3

SIMULATE a sample of 1,000 draws from the posterior model of a, b, and s. Store this mcmc.list as weight_sim.

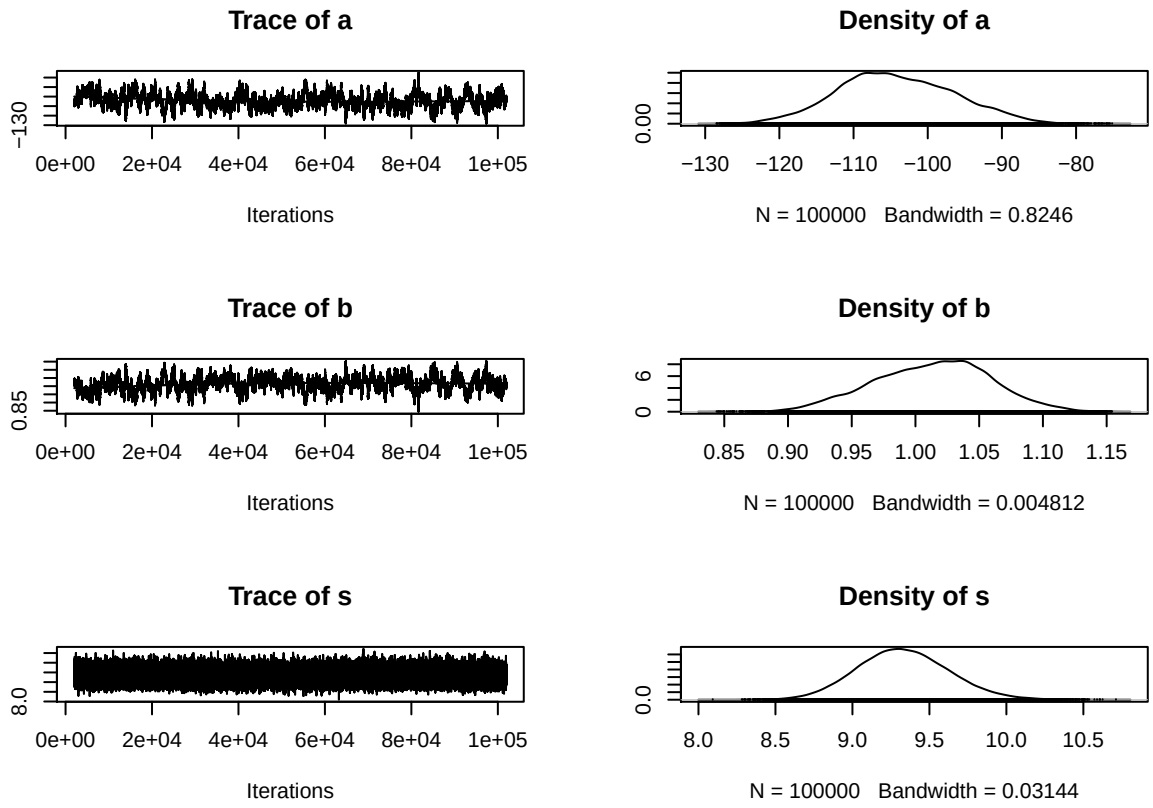
Use plot() to construct trace and density plots of the posterior samples in weight_sim.

```
# SIMULATE the posterior
weight_sim <- coda.samples(model = weight_jags, variable.names = c("a", "b", "s"), n.iter = 1000)

# PLOT the posterior
plot(weight_sim)
```



The results don't look that great. Let's fix that by increasing the number of simulations.

```
# SIMULATE the posterior
weight_sim_big <- coda.samples(model = weight_jags, variable.names = c("a", "b", "s"), n.iter = 100000)

# PLOT the posterior
plot(weight_sim_big)
```



Trace plots indicate that after only 1,000 iterations, the a and b parallel chains had not stabilized. However, after 100,000 iterations, the chains demonstrate greater stability. We might also increase the stability of our simulation by standardizing the height data.

Posterior point estimates

Recall the likelihood of the Bayesian regression model of weight Y by height X: $Y_i \sim N(m_i, s^2)$ where $m_i = a + bX_i$. Load a dataframe with the contents of `weight_sim_big` markov chain output:

```
weight_chains = data.frame(as.matrix(weight_sim_big[1]), iter = 1:100000)
```

The posterior means of the intercept & slope parameters, a & b, reflect the posterior mean trend in the relationship between weight & height. In contrast, the full posteriors of a & b reflect the range of plausible parameters, thus posterior uncertainty in the trend. You will examine the trend and uncertainty in this trend below.

Instructions

- Obtain `summary()` statistics of the `weight_sim_big` chains.
- The posterior mean of b is reported in Table 1 of the `summary()`. Use the raw `weight_chains` to verify this calculation.
- Construct a scatterplot of the `wgt` vs `hgt` data in `bdims`. Use `geom_abline()` to superimpose the posterior mean trend.
- Construct another scatterplot of `wgt` vs `hgt`. Superimpose the 20 regression lines defined by the first 20 sets of `a` & `b` parameter values in `weight_chains`.

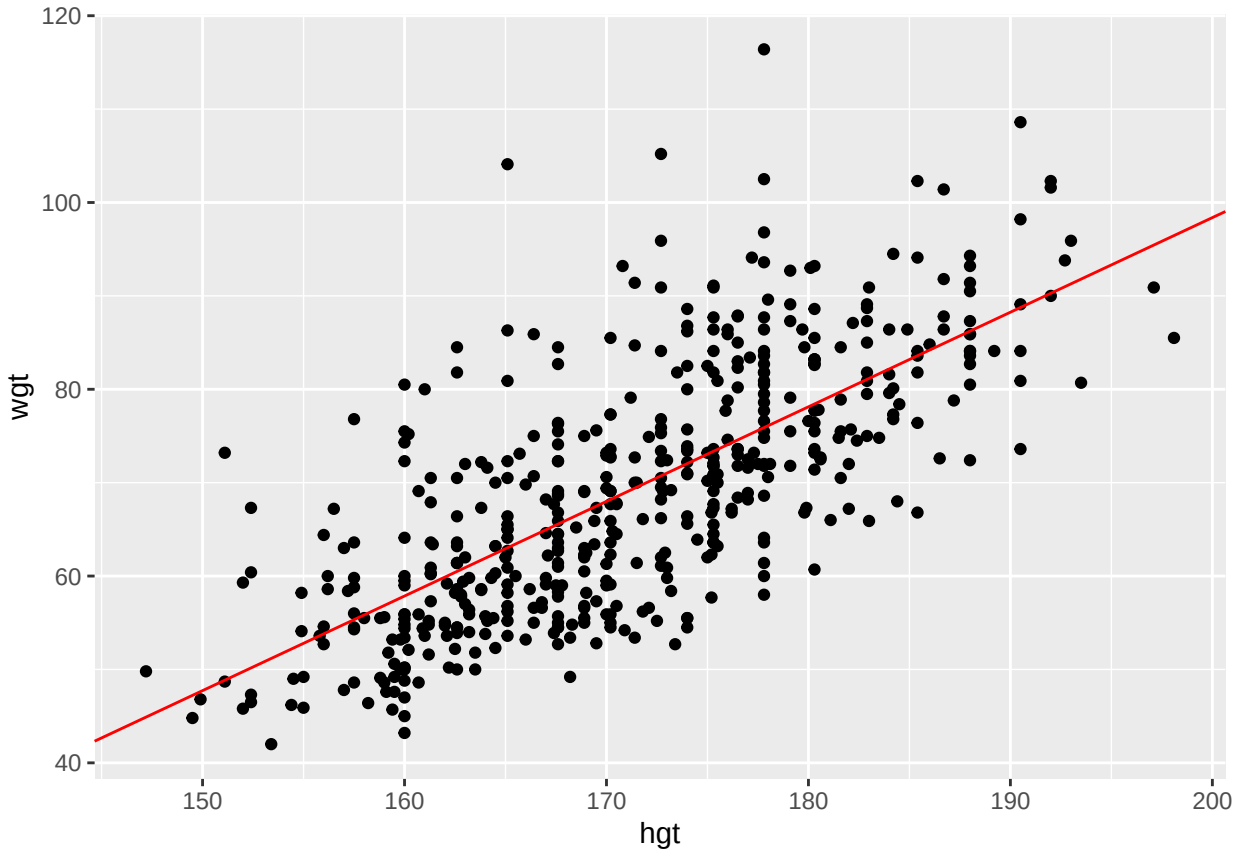
```r
# Summarize the posterior Markov chains
summary(weight_sim_big)
```

```
##
## Iterations = 2001:102000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1e+05
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##       Mean      SD  Naive SE Time-series SE
## a -104.222 7.77968 0.0246015       0.646878
## b    1.013 0.04539 0.0001435       0.003794
## s    9.331 0.29656 0.0009378       0.001214
##
## 2. Quantiles for each variable:
##
##        2.5%       25%      50%     75%   97.5%
## a -118.8879 -109.7096 -104.681 -98.780 -88.712
## b    0.9224    0.9813    1.016   1.045   1.098
## s    8.7743    9.1273    9.323   9.526   9.938
```

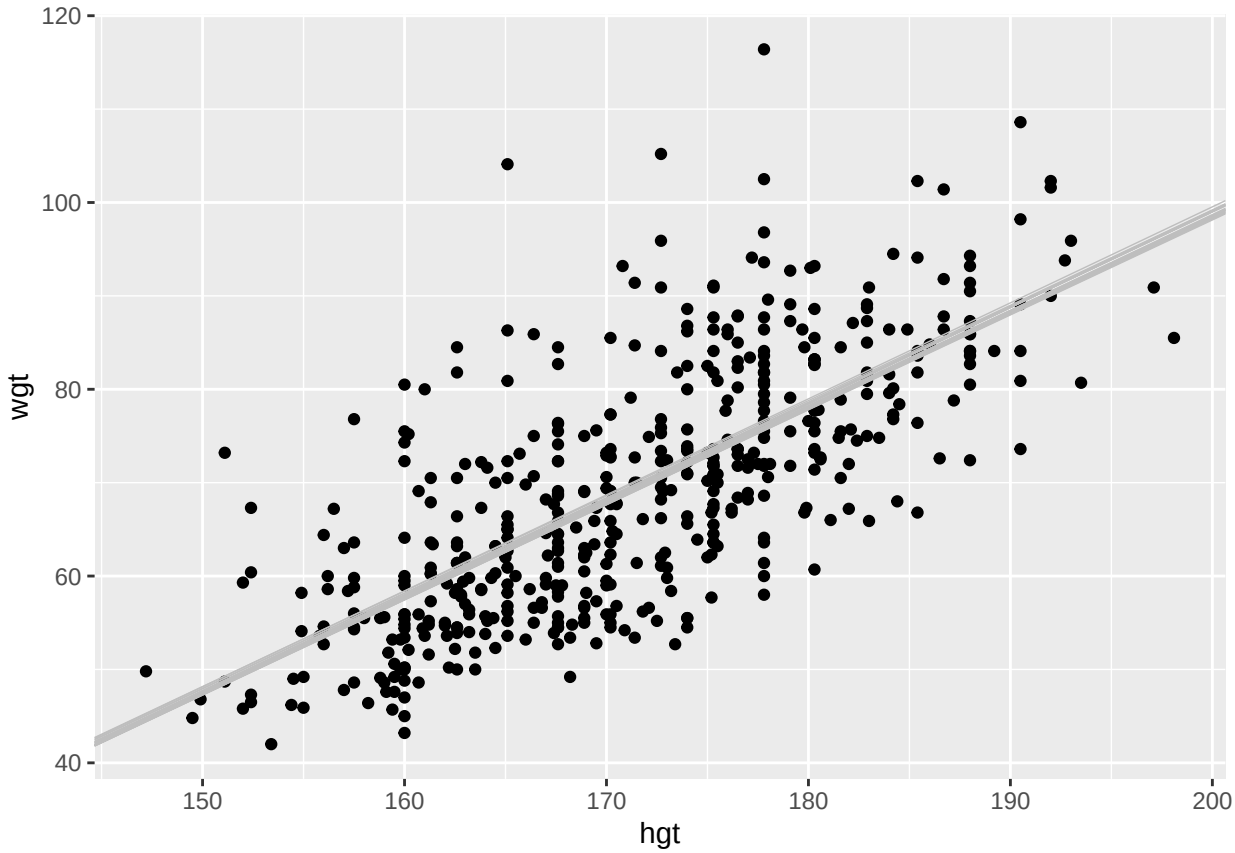
```r
# Calculate the estimated posterior mean of b
mean(weight_chains$b)
```

```
## [1] 1.013019
```

```r
# Plot the posterior mean regression model
ggplot(bdims, aes(x = hgt, y = wgt)) +
    geom_point() +
    geom_abline(intercept = mean(weight_chains$a), slope = mean(weight_chains$b), color = "red")
```




```
# Visualize the range of 20 posterior regression models
ggplot(bdims, aes(x = hgt, y = wgt)) +
    geom_point() +
    geom_abline(intercept = weight_chains$a[1:20], slope = weight_chains$b[1:20], color = "gray", size =
```



Given the size of the data and selection of priors, the posterior uncertainty is noticeably small as evidenced by the tight distribution of the gray posterior plausible lines around the trend.

Posterior credible intervals

Let's focus on slope parameter b, the rate of change in weight over height. The posterior mean of b reflects the trend in the posterior model of the slope. In contrast, a posterior credible interval provides a range of posterior plausible slope values, thus reflects posterior uncertainty about b. For example, the 95% credible interval for b ranges from the 2.5th to the 97.5th quantile of the b posterior. Thus there's a 95% (posterior) chance that b is in this range.

You will use RJAGS simulation output to approximate credible intervals for b.

Instructions

- Obtain `summary()` statistics of the weight_sim_big chains.
- The 2.5% and 97.5% posterior quantiles for b are reported in Table 2 of the `summary()`. Apply `quantile()` to the raw `weight_chains` to verify these calculations. Save this as `ci_95` and print it.
- Similarly, use the `weight_chains` data to construct a 90% credible interval for b. Save this as `ci_90` and print it.
- Construct a density plot of the b Markov chain values. Superimpose vertical lines representing the 90% credible interval for b using `geom_vline()` with `xintercept = ci_90`.

```
# Summarize the posterior Markov chains
summary(weight_sim_big)
```

```
##
## Iterations = 2001:102000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1e+05
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##       Mean      SD  Naive SE Time-series SE
## a -104.222 7.77968 0.0246015       0.646878
## b    1.013 0.04539 0.0001435       0.003794
## s    9.331 0.29656 0.0009378       0.001214
##
## 2. Quantiles for each variable:
##
##        2.5%       25%      50%     75%   97.5%
## a -118.8879 -109.7096 -104.681 -98.780 -88.712
## b    0.9224    0.9813    1.016   1.045   1.098
## s    8.7743    9.1273    9.323   9.526   9.938
```

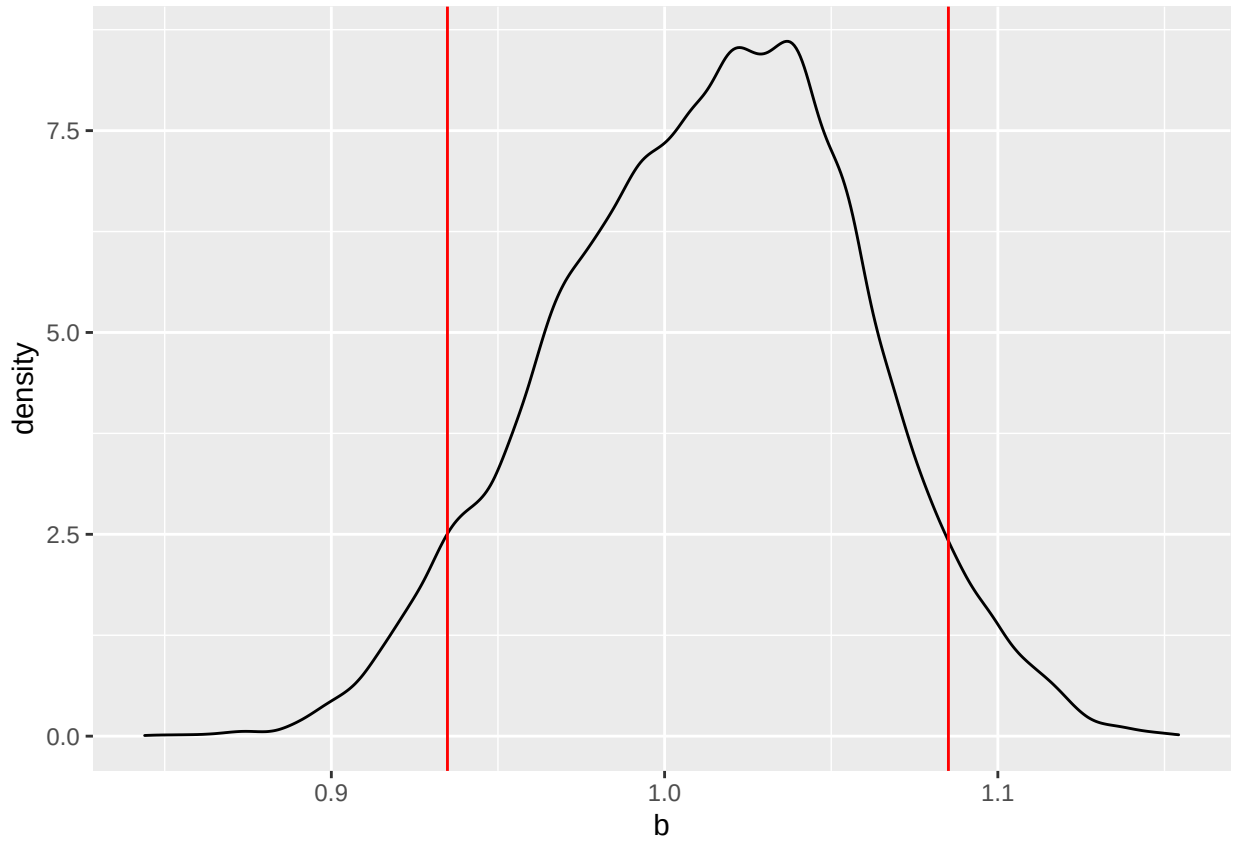
```
# Calculate the 95% posterior credible interval for b
ci_95 <- quantile(weight_chains$b, probs = c(0.025, 0.975))
ci_95
```

```
##      2.5%     97.5%
## 0.9224311 1.0983272
```

```
# Calculate the 90% posterior credible interval for b
ci_90 <- quantile(weight_chains$b, probs = c(0.05, 0.95))
ci_90
```

```
##        5%       95%
## 0.9348413 1.0851638
```

```
# Mark the 90% credible interval
ggplot(weight_chains, aes(x = b)) +
    geom_density() +
    geom_vline(xintercept = ci_90, color = "red")
```



Based on your calculations we can say that there's a 90% (posterior) probability that, on average, the increase in weight per 1 cm increase in height is between 0.93 and 1.08 kg.

Posterior probabilities

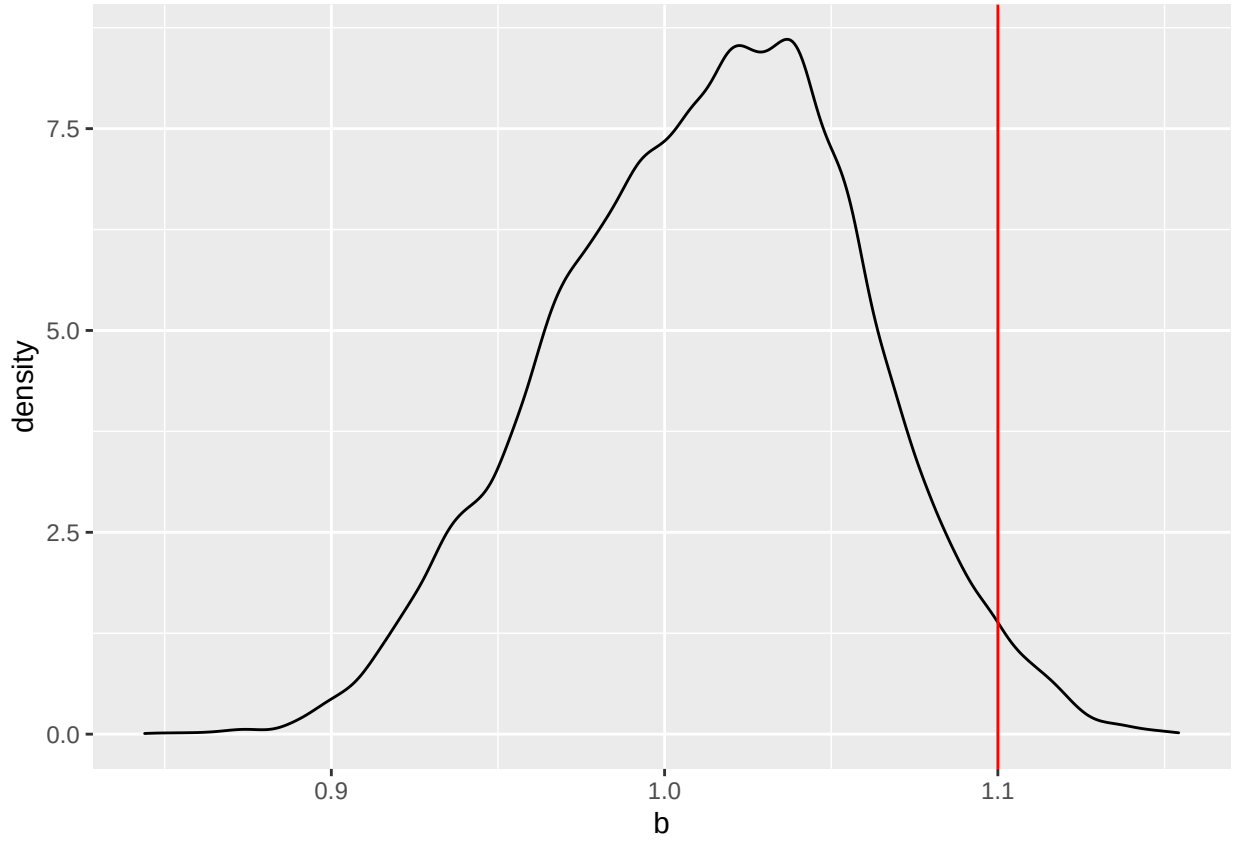
You've used RJAGS output to explore and quantify the posterior trend & uncertainty b. You can also use RJAGS output to assess specific hypotheses. For example: What's the posterior probability that, on average, weight increases by more than 1.1 kg for every 1 cm increase in height? That is, what's the posterior probability that b>1.1?

You will approximate this probability by the proportion of b Markov chain values that exceed 1.1.

Instructions

- Construct a density plot of the b Markov chain values and use `geom_vline()` to superimpose a vertical line at 1.1.
- Use `table()` to summarize the number of b Markov chain values that exceed 1.1.
- Use `mean()` to calculate the proportion of b Markov chain values that exceed 1.1.

```
# Mark 1.1 on a posterior density plot for b
ggplot(weight_chains, aes(x = b)) +
    geom_density() +
    geom_vline(xintercept = 1.1, color = "red")
```




```
# Summarize the number of b chain values that exceed 1.1
table(weight_chains$b>1.1)
```

```
##
## FALSE  TRUE
## 97751  2249
```

```
# Calculate the proportion of b chain values that exceed 1.1
mean(weight_chains$b>1.1)
```

```
## [1] 0.02249
```

Based on the abve calculations we can say that there's only a ~2% (posterior) chance that, on average, the increase in weight per 1 cm increase in height exceeds 1.1 kg.

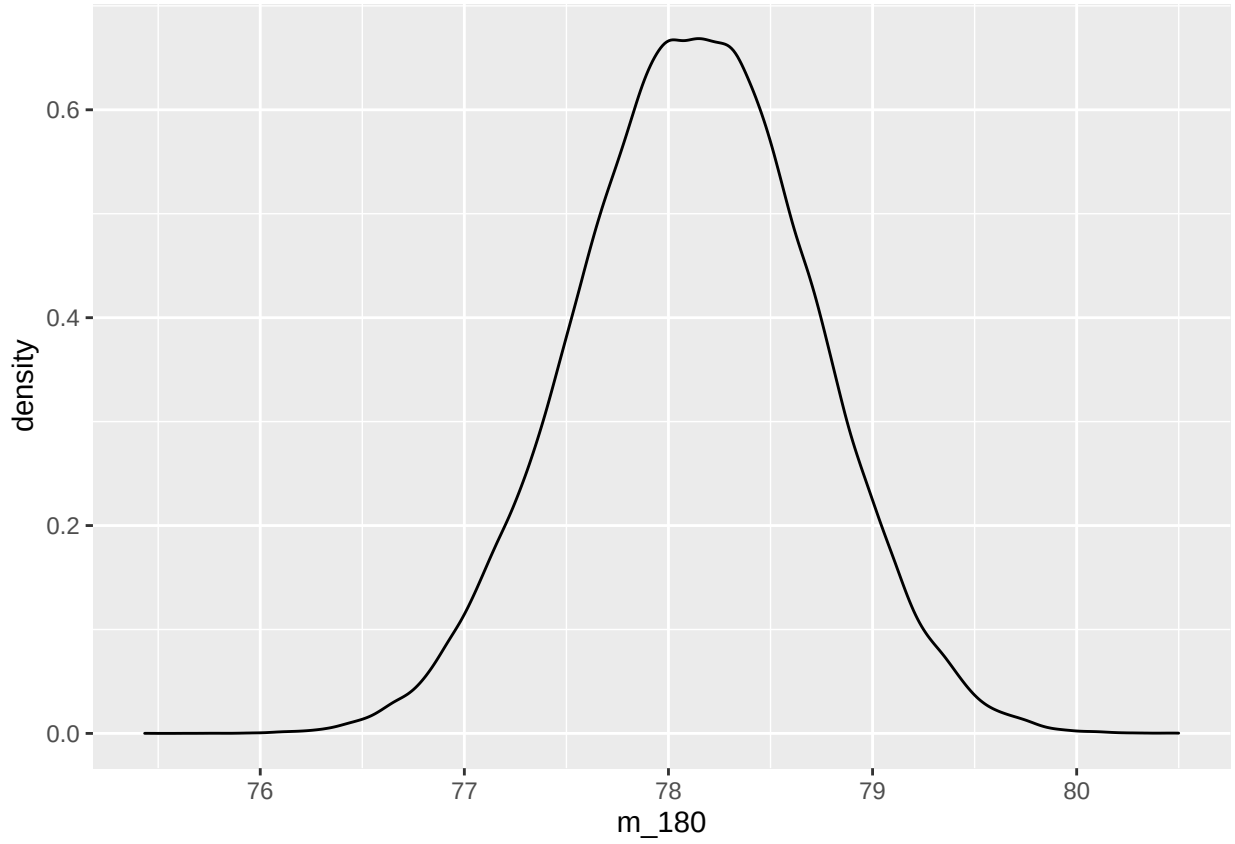
Inference for the posterior trend

In earlier exercises you approximated the form of the posterior trend m (solid line). From this, notice that the typical weight among 180 cm adults is roughly 80 kg. You will use RJAGS simulation output to approximate the posterior trend in weight among 180 cm tall adults as well as the posterior uncertainty in this trend.

Instructions

- `weight_chains` contains 100,000 sets of posterior plausible parameter values of a and b. From each, calculate the mean (typical) weight among 180 cm tall adults, `a + b * 180`. Store these trends as a new variable `m_180` in `weight_chains`.
- Construct a posterior density plot of 100,000 `m_180` values.
- Use the 100,000 `m_180` values to calculate a 95% posterior credible interval for the mean weight among 180 cm tall adults.

```
# Calculate the trend under each Markov chain parameter set
weight_chains <- weight_chains  %>%
    mutate(m_180 = a + b*180)

# Construct a posterior density plot of the trend
ggplot(weight_chains, aes(x = m_180)) +
    geom_density()
```




```
# Construct a posterior credible interval for the trend
quantile(weight_chains$m_180, probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 76.97793 79.23810
```

The posterior trend of your regression model indicates that the typical weight among 180 cm tall adults is roughly 78 kg. However, posterior uncertainty in the regression model trickles down to uncertainty about this trend. This uncertainty is communicated through your credible interval: there's a 95% (posterior) chance that the typical weight at a height of 180 cm is between 76.95 and 79.24 kg.

Calculating posterior predictions

You just explored the posterior trend in weight Y among adults with height X=180: $m_{180} = a + b * 180$. What if you wanted to predict the weight of a specific 180 cm tall adult? You can! To do so, you must account for individual variability from the trend, modeled by

$Y_{180} \sim N(m_{180}, s^2)$

Using this model, you will simulate predictions of weight under each set of posterior plausible parameters in `weight_chains`.

Instructions

Use `rnorm()` to simulate a single prediction of weight under the parameter settings in the first row of `weight_chains`. Repeat the above using the parameter settings in the second row of `weight_chains`. Simulate a single prediction of weight under each of the 100,000 parameter settings in `weight_chains`. Store these as a new variable `Y_180` in `weight_chains`. Print the first 6 rows of parameter values & predictions in `weight_chains`.

```
# Simulate 1 prediction under the first parameter set
rnorm(n = 1, mean = weight_chains$m_180[1], sd = weight_chains$s[1])
```

```
## [1] 81.19463
```

```
# Simulate 1 prediction under the second parameter set
rnorm(n = 1, mean = weight_chains$m_180[2], sd = weight_chains$s[2])
```

```
## [1] 75.33444
```

```
# Simulate & store 1 prediction under each parameter set
weight_chains <- weight_chains  %>%
    mutate(Y_180 = rnorm(n = 100000, mean = weight_chains$m_180, sd = weight_chains$s))

# Print the first 6 parameter sets & predictions
head(weight_chains)
```

```
##           a        b        s iter    m_180    Y_180
## 1 -106.5042 1.027620 9.053909    1 78.46745 77.54361
## 2 -106.6708 1.029252 9.159864    2 78.59459 74.50711
## 3 -106.5268 1.026147 9.441036    3 78.17955 76.80833
## 4 -106.1365 1.023241 9.607720    4 78.04680 86.61513
## 5 -106.0383 1.027861 9.274800    5 78.97676 80.97807
## 6 -107.2631 1.031626 9.155143    6 78.42966 92.53365
```

You will use these 100,000 predictions to approximate the posterior predictive distribution for the weight of a 180 cm tall adult.

Instructions

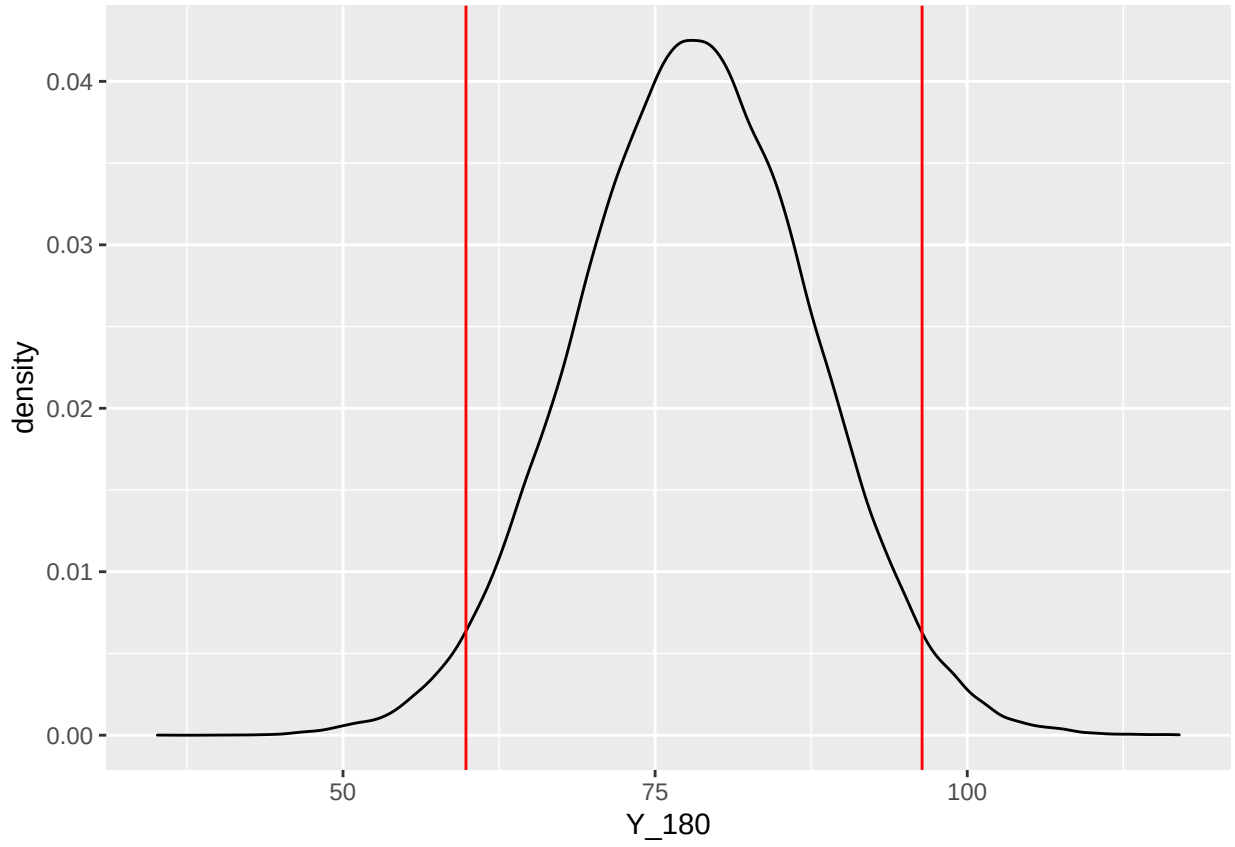
- Use the 10,000 `Y_180` values to construct a 95% posterior credible interval for the weight of a 180 cm tall adult.
- Construct a density plot of your 100,000 posterior plausible predictions.
- Construct a scatterplot of the `wgt` vs `hgt` data in `bdims`.
  - Use `geom_abline()` to superimpose the posterior regression trend.

– Use `geom_segment()` to superimpose a vertical line at a hgt of 180 that represents the lower & upper limits (`y` and `yend`) of `ci_180`.

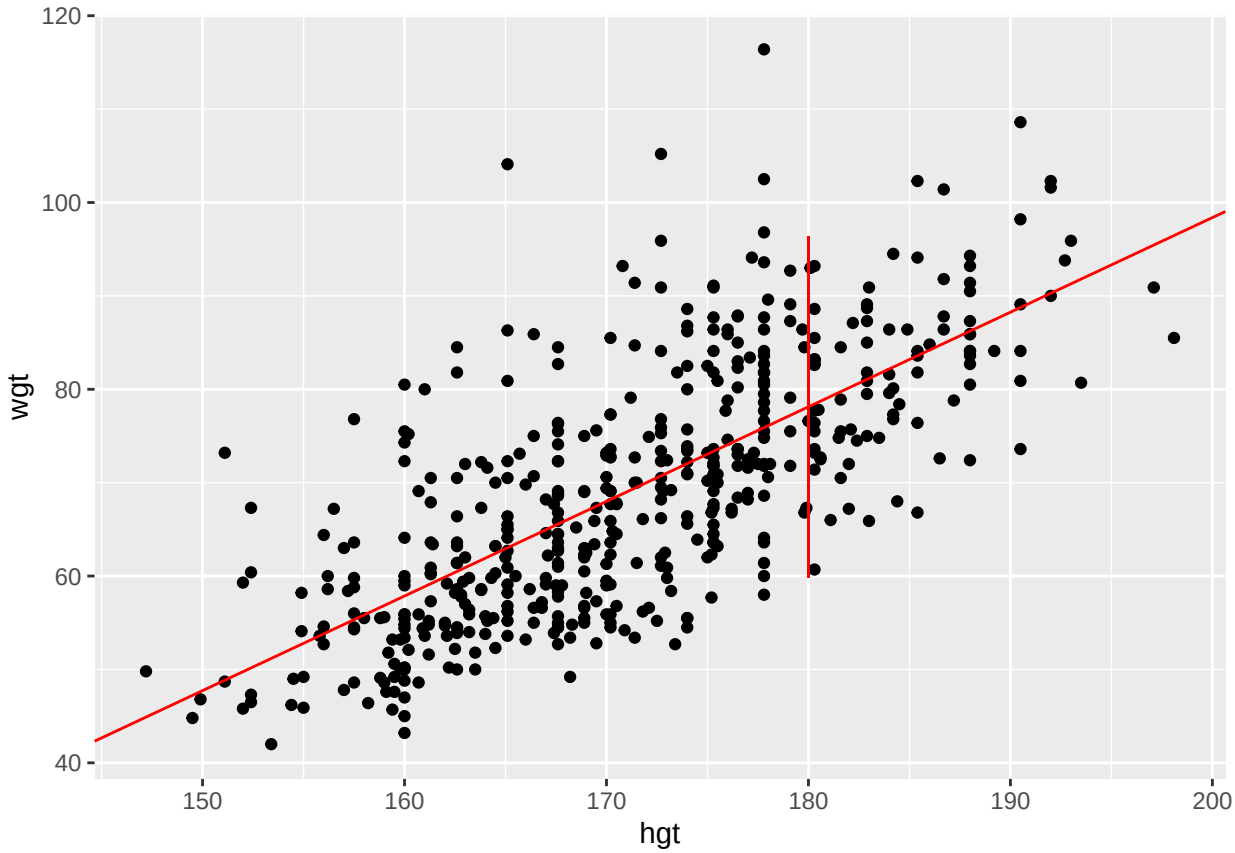
```
# Construct a posterior credible interval for the prediction
ci_180 <- quantile(weight_chains$Y_180, probs = c(0.025, 0.975))
ci_180
```

```
##     2.5%    97.5%
## 59.85000 96.38405
```

```
# Construct a density plot of the posterior predictions
ggplot(weight_chains, aes(x = Y_180)) +
    geom_density() +
    geom_vline(xintercept = ci_180, color = "red")
```




```
# Visualize the credible interval on a scatterplot of the data
ggplot(bdims, aes(x = hgt, y = wgt)) +
    geom_point() +
    geom_abline(intercept = mean(weight_chains$a), slope = mean(weight_chains$b), color = "red") +
    geom_segment(x = 180, xend = 180, y = ci_180[1], yend = ci_180[2], color = "red")
```



Congratulations! You've simulated your first posterior predictive distribution. Your 100,000 posterior plausible weights for a given 180 cm tall adult ranged from roughly 36 to 117 kg. Eliminating the most extreme 5% of these predictions, you observed that there's a 95% (posterior) chance that the weight is between 72 and 84 kg.