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

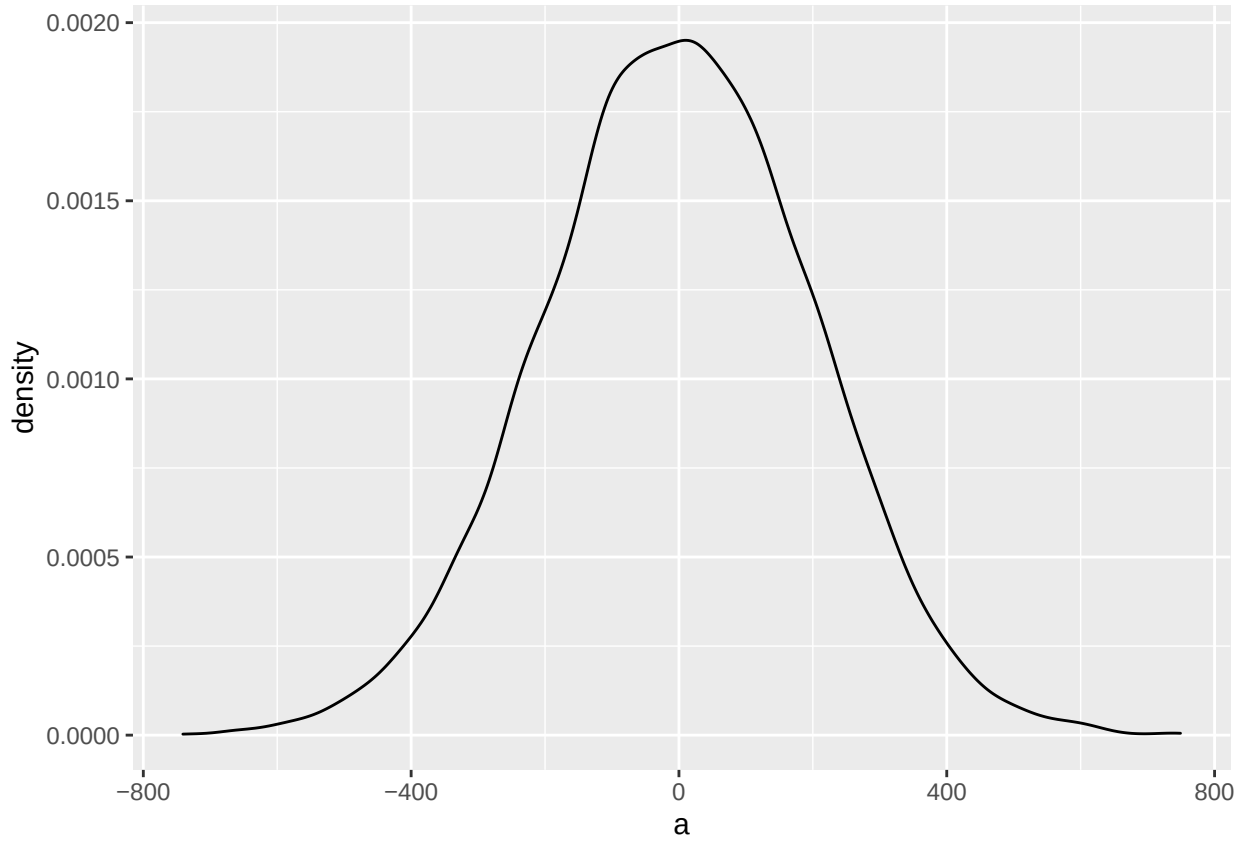
Let Y_i be the weight (in kg) of subject i. Past studies have shown that weight is linearly related to height X_i (in cm). The average weight m_i among adults of any shared height X_i can be written as $m_i = a + bX_i$. But height isn't a perfect predictor of weight - individuals vary from the trend. 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 $c \sim Unif(0, 20)$.

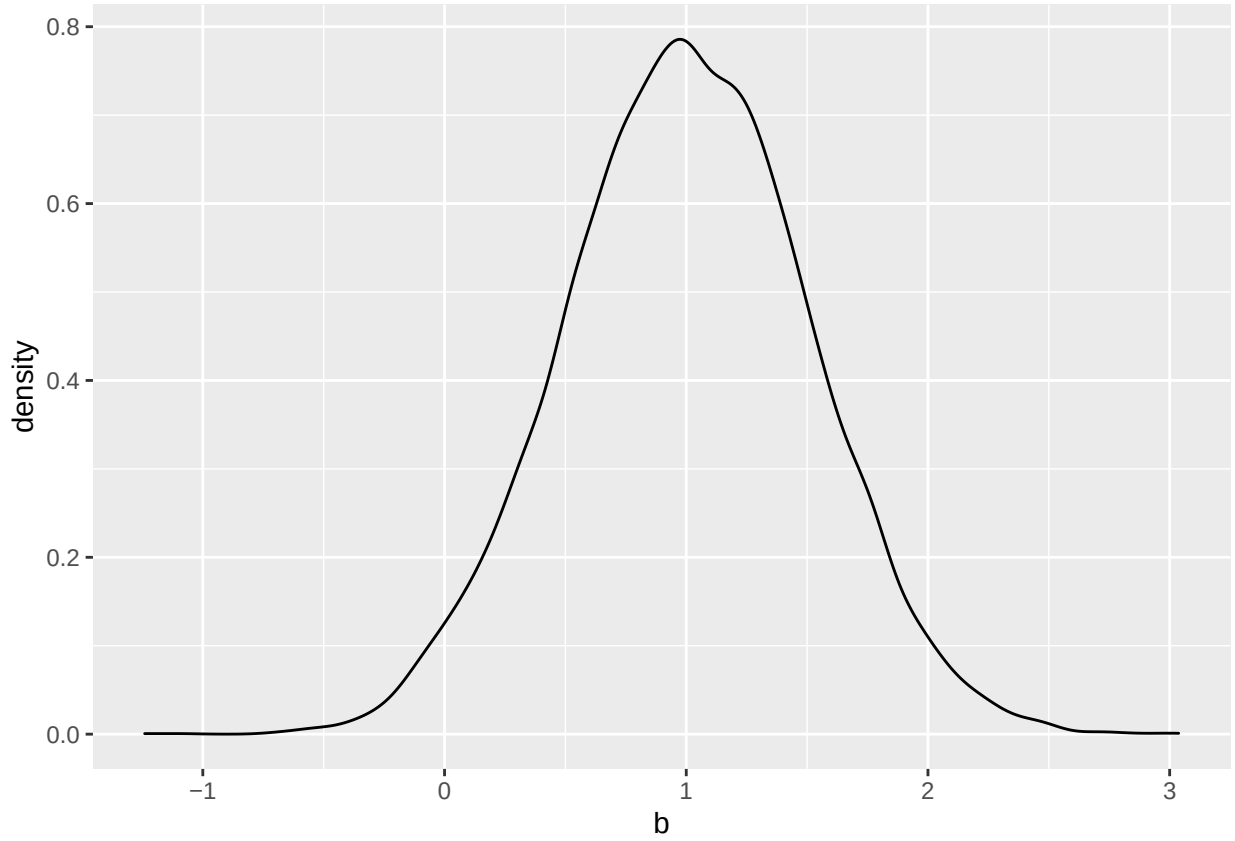
Instructions

- 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.

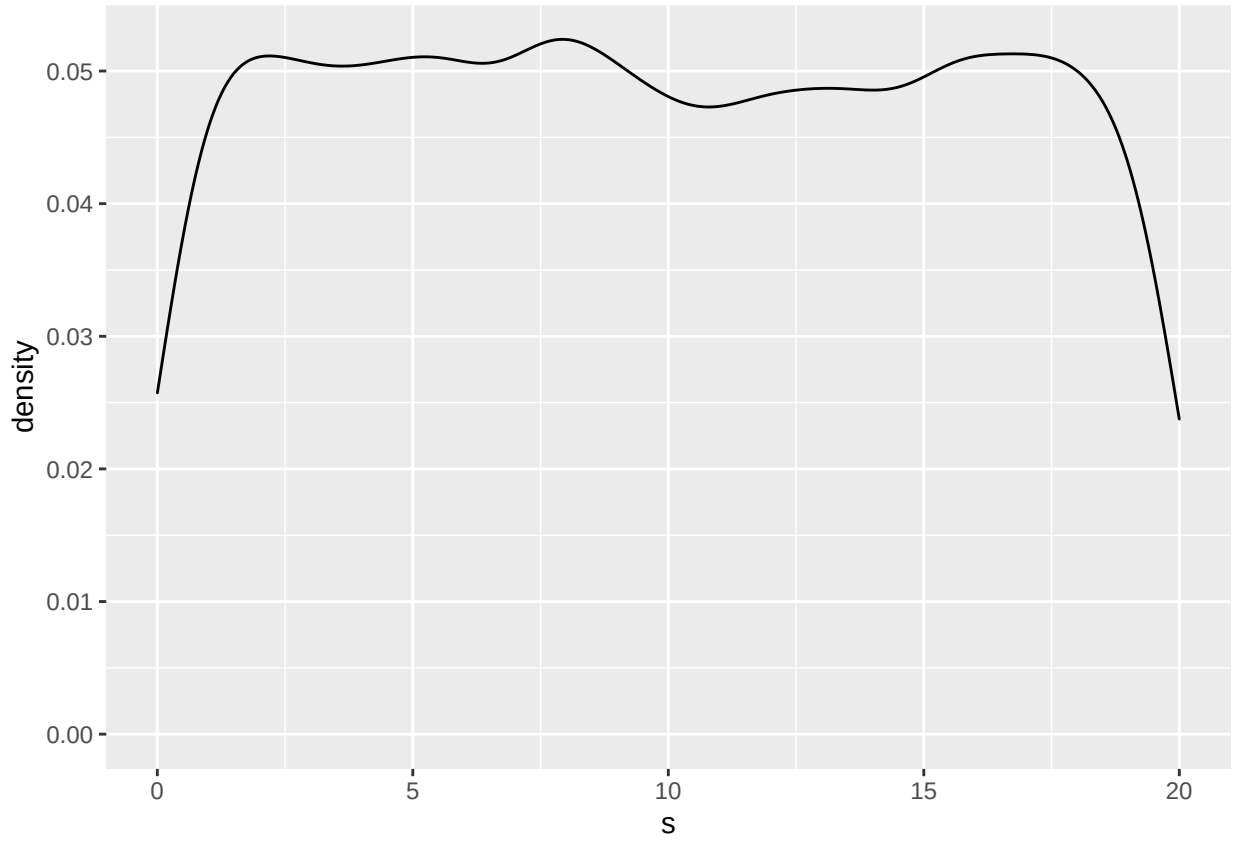
```
# Take 10000 samples from the a, b, & s priors
a <- rnorm(10000,0,200)
b <- rnorm(10000,1,0.5)
s <- runif(10000,0,20)

# 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

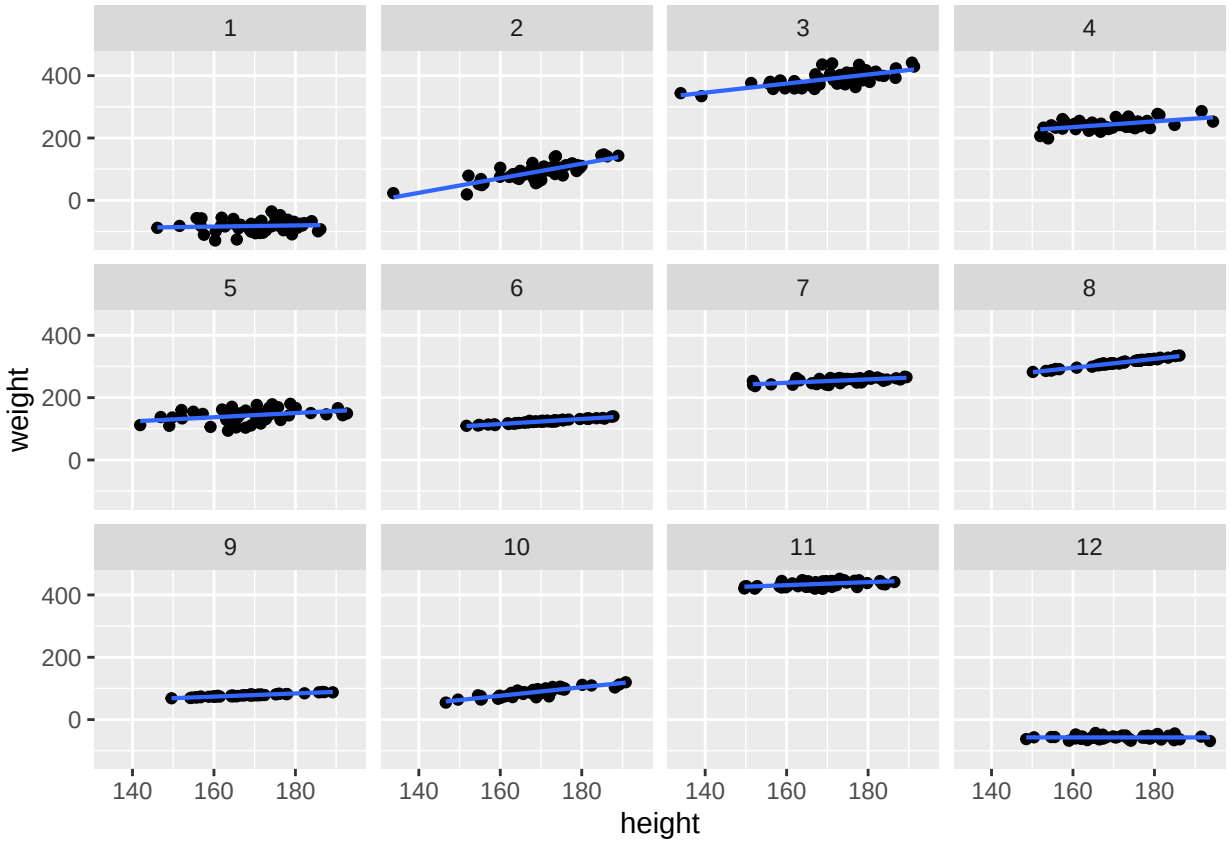
In the previous exercise, you simulated 10,000 samples for each parameter (a, b, s) in the Bayesian regression model of weight Y by height X: $Y_i \sim N(m_i, s^2)$ with mean $m = a + bX$. The set of a, b, and s values in each row of samples represents a prior plausible regression scenario. To explore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

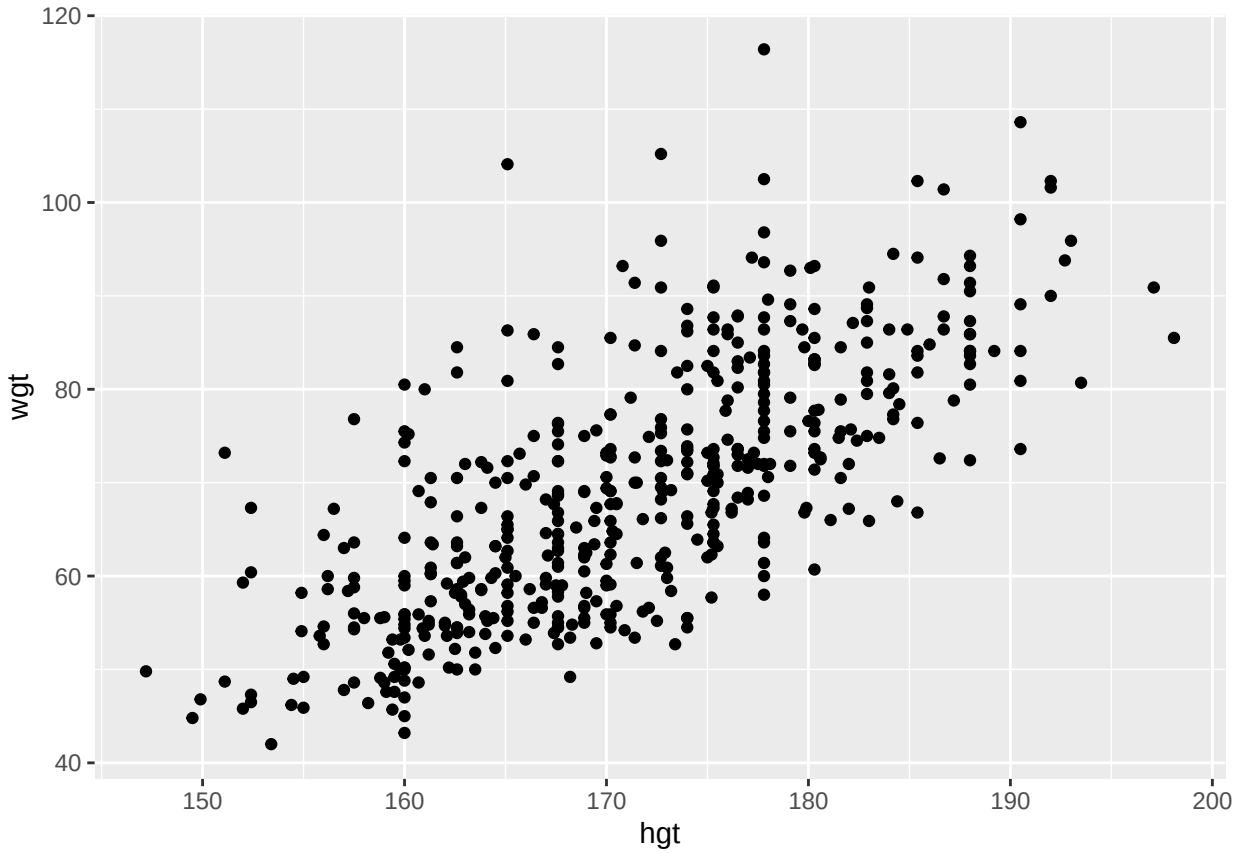
```
## The following objects are masked from 'package:datasets':
##
##     cars, trees
```

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

instructions

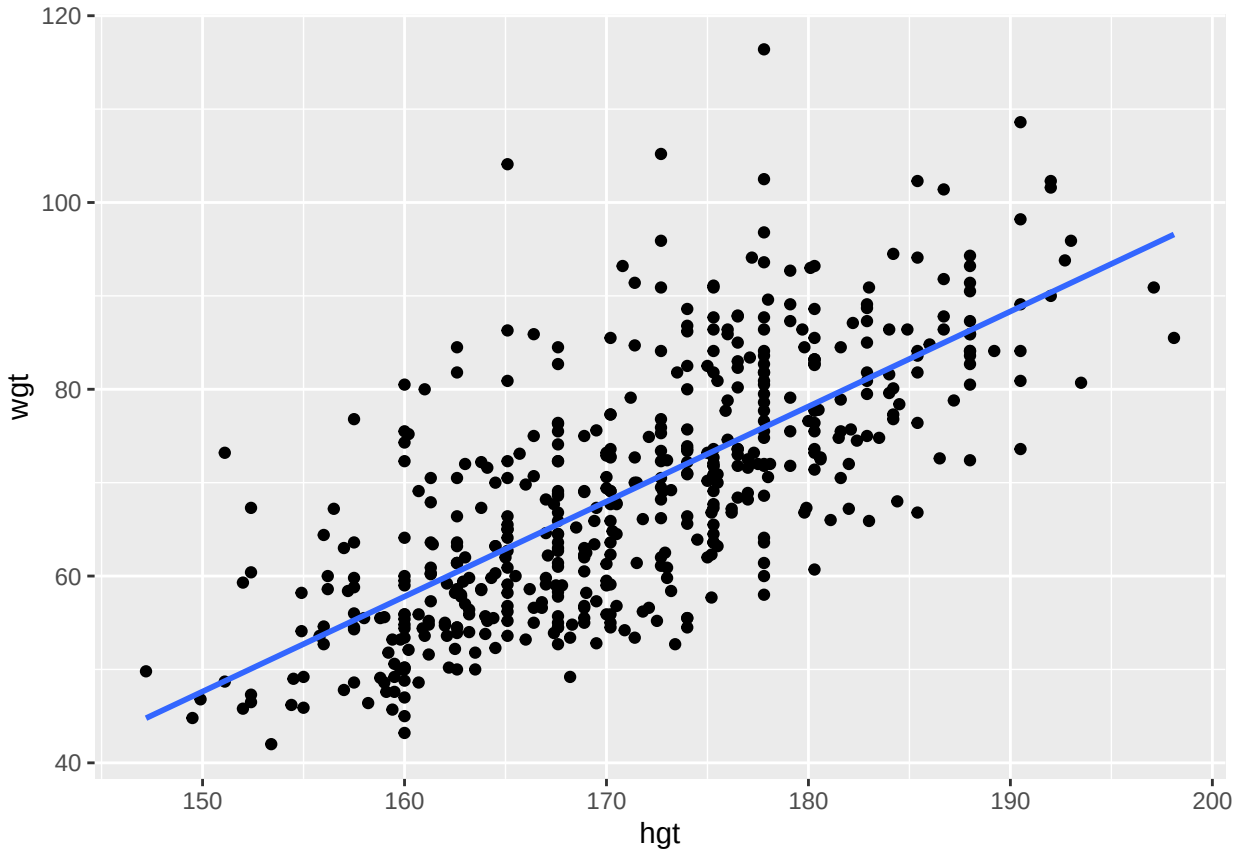
- Construct a scatterplot of `wgt` (y-axis) vs `hgt` (x-axis) using `ggplot()` with a `geom_point()` layer.
- Construct a scatterplot of wgt vs hgt which includes a `geom_smooth()` of the linear relationship between these 2 variables.

```
# 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)
)
```

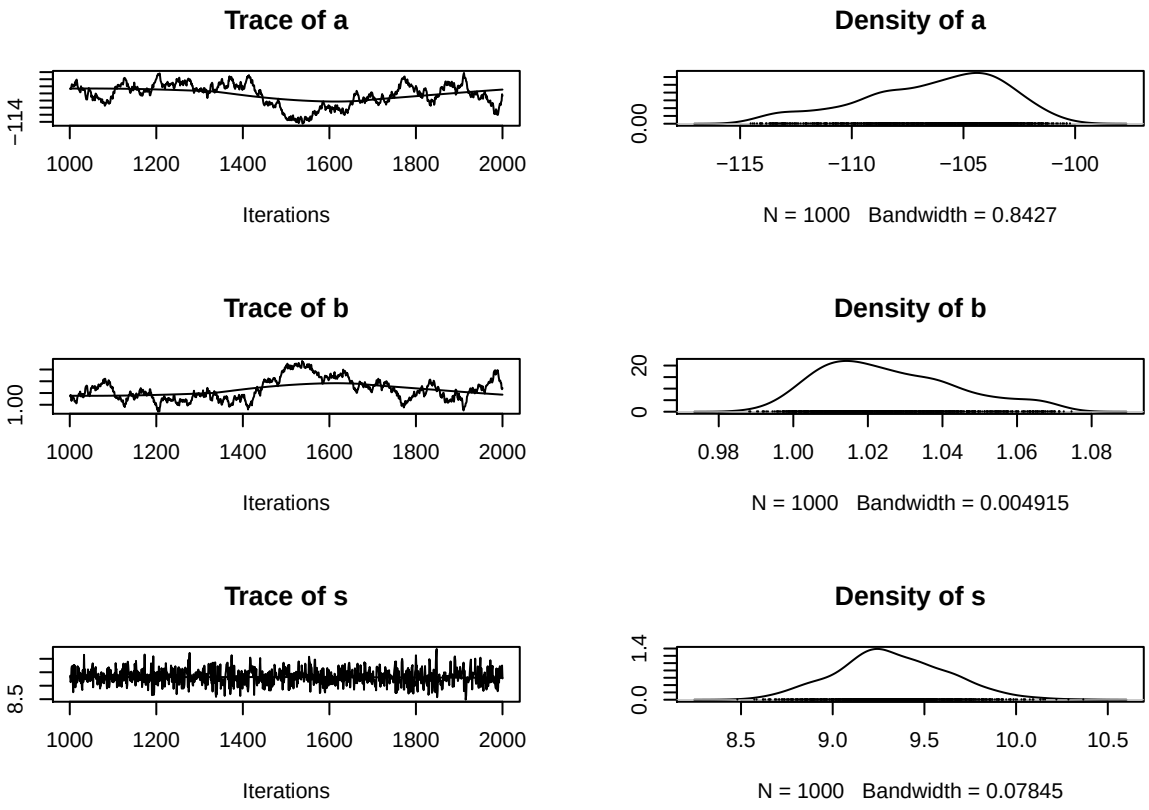
```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 507
##    Unobserved stochastic nodes: 3
##    Total graph size: 1321
##
## Initializing model
```

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.

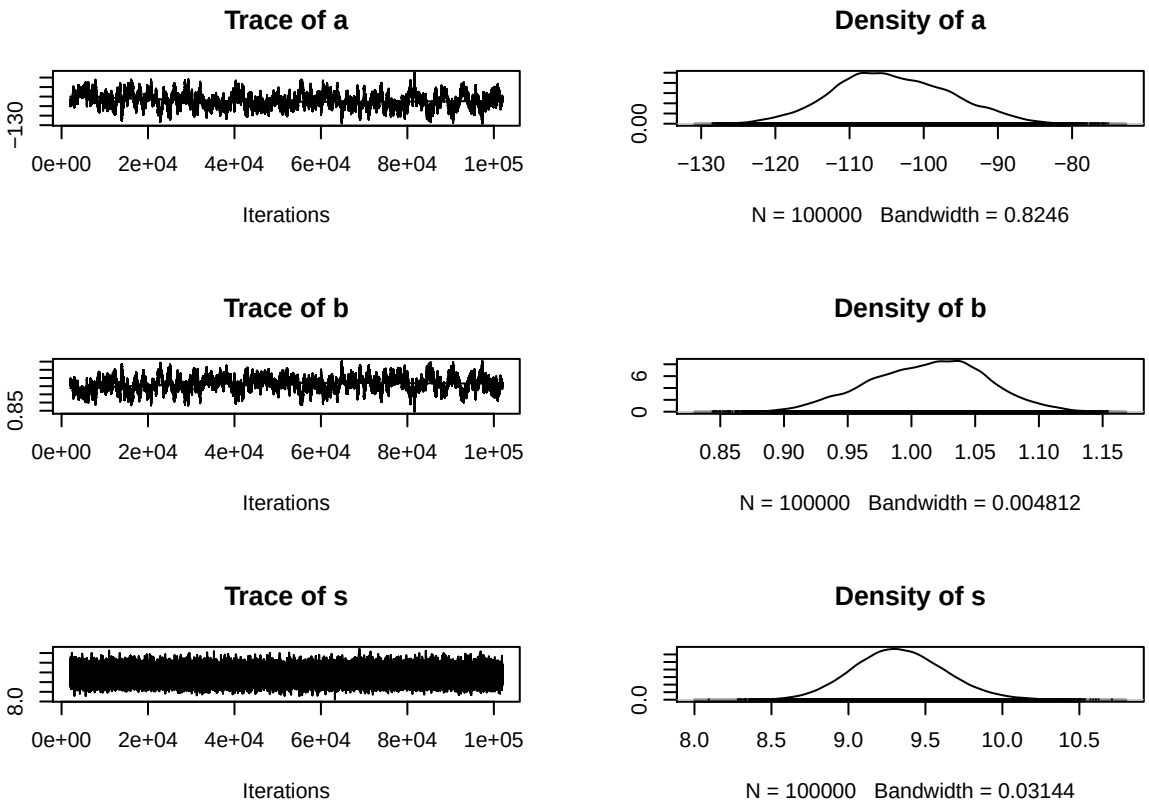
Use plot() to construct trace and density plots of the posterior samples in weight_sim.

```
# 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.

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

# PLOT the posterior
plot(weight_sim)
```



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.