

Easy

E1

$y_i \sim \text{Normal}(\mu, \sigma)$ is the likelihood.

E2

There are two parameters: μ and σ .

E3

$$[\mu, \sigma \mid y] \propto \prod_i^n \text{Normal}(y_i \mid \mu, \sigma) \times \\ \text{Normal}(\mu \mid 0, 10) \times \\ \text{Uniform}(\sigma \mid 0, 10)$$

E4

$\mu_i = \alpha + \beta x_i$ is the linear model.

E5

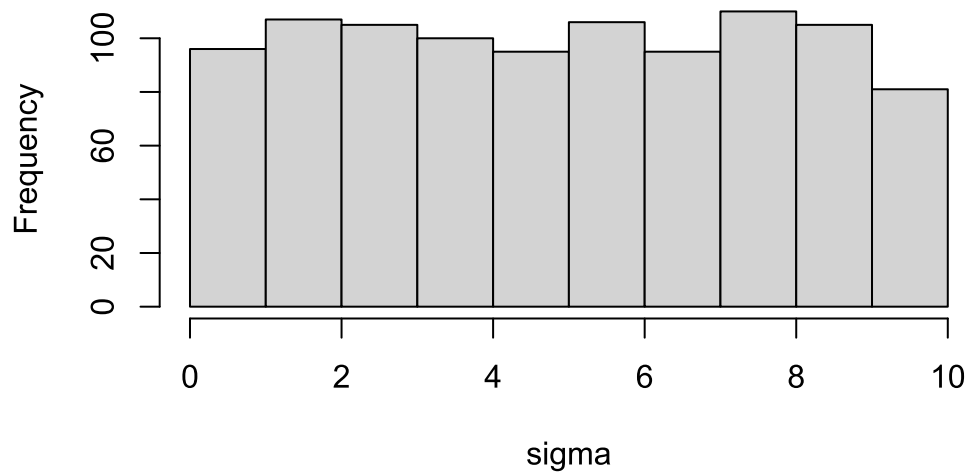
There are 3 parameters: α , β , and σ .

Medium

M1

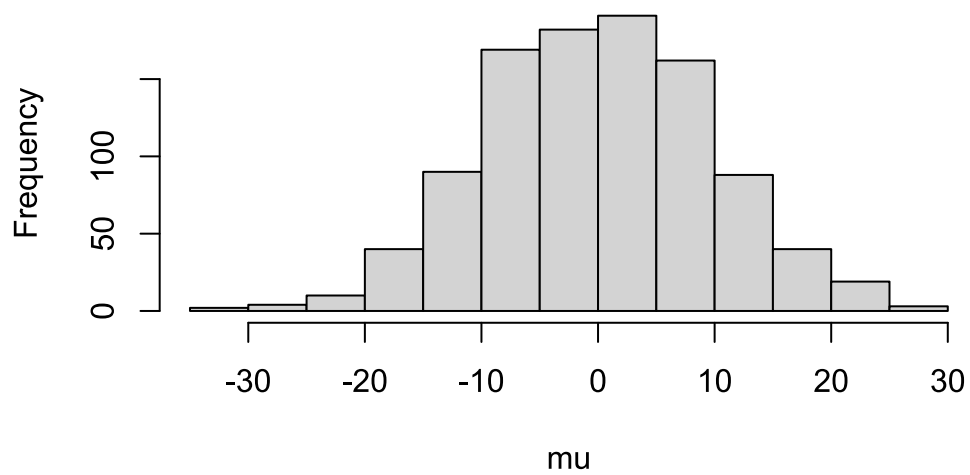
```
sigma <- runif(1000, 0, 10)
mu <- rnorm(1000, 0, 10)
y <- rnorm(mu, sigma)
hist(sigma)
```

Histogram of sigma

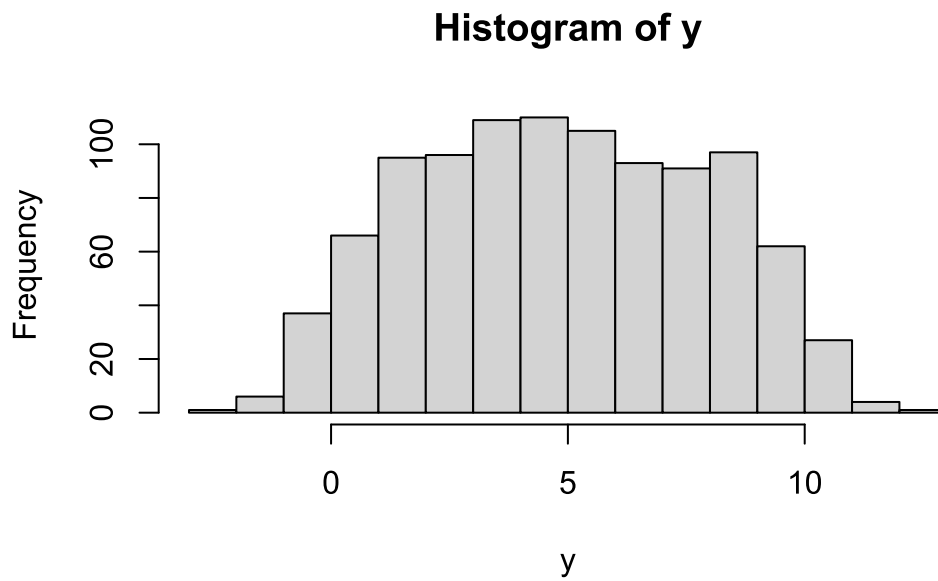


```
hist(mu)
```

Histogram of mu



```
hist(y)
```



M2

```
alist(
  y ~ dnorm(mu, sigma),
  mu ~ dnorm(0, 10),
  sigma ~ dunif(0,10)
)
```

```
[[1]]
y ~ dnorm(mu, sigma)
```

```
[[2]]
mu ~ dnorm(0, 10)
```

```
[[3]]
sigma ~ dunif(0, 10)
```

M3

$$y_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta x_i$$

$$\alpha \sim \text{Normal}(0, 50)$$

$$\beta \sim \text{Uniform}(0, 10)$$

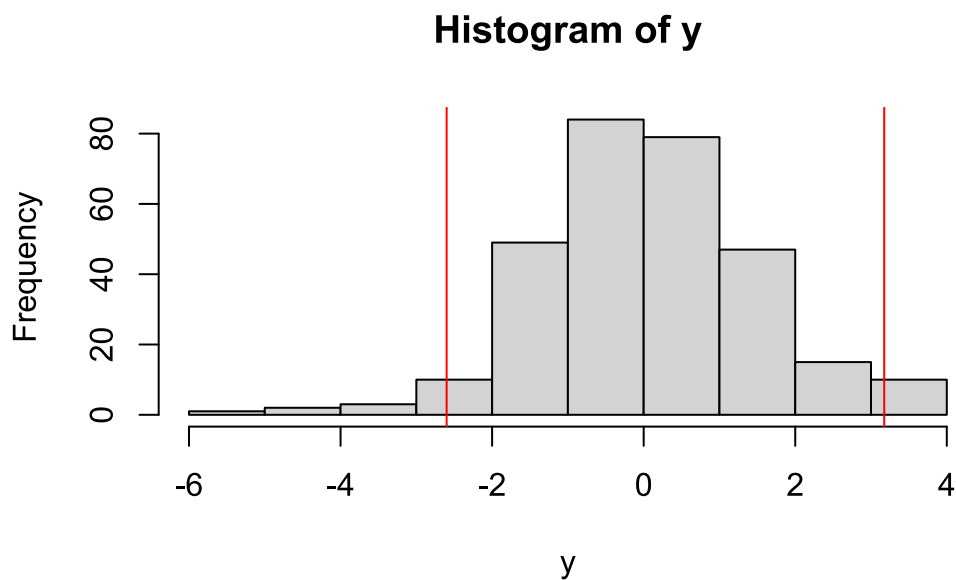
$$\sigma \sim \text{Uniform}(0, 50)$$

M4

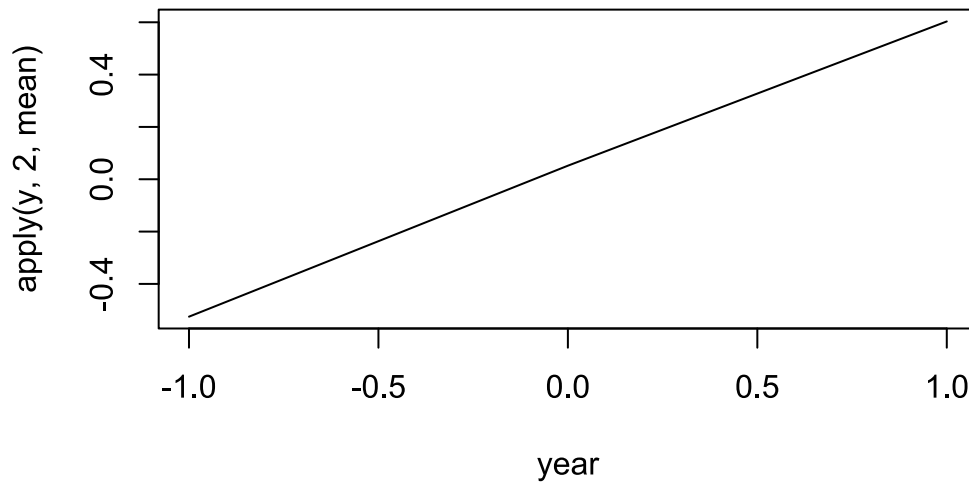
I will standardize height across all three years to make choosing priors easier. I will also code the first year as -1 , the second as 0 and the third year as 1 . This will make α very close to zero. Setting β to likely be positive (kids tend to grow). And $\log(\sigma)$ has a distribution around 1 . These give a prior predictive distribution that seems reasonable

```
# center year so that it is (-1, 0, 1)
# standardize height measurements so that they have a mean of 0 and a stdv of 1
n <- 100
a <- rnorm(n, 0, .5)
b <- rnorm(n, .5, .5)
y <- matrix(nrow = n, ncol = 3)
sigma <- rnorm(n, 0, .25)
for(i in 1:3){
  y[,i] <- rnorm(n, a + b*(i-2), exp(sigma))
}

hist(y)
abline(v = quantile(y, c(.025, .975)),
       col = "red")
```



```
year <- c(-1, 0, 1)
plot(apply(y, 2, mean) ~ year, type = "l")
```



$$height_i \sim Normal(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta \times year_i$$

$$\alpha \sim Normal(0, .5)$$

$$\beta \sim Normal(.5, .5)$$

$$\log(\sigma) \sim Normal(0, .25)$$

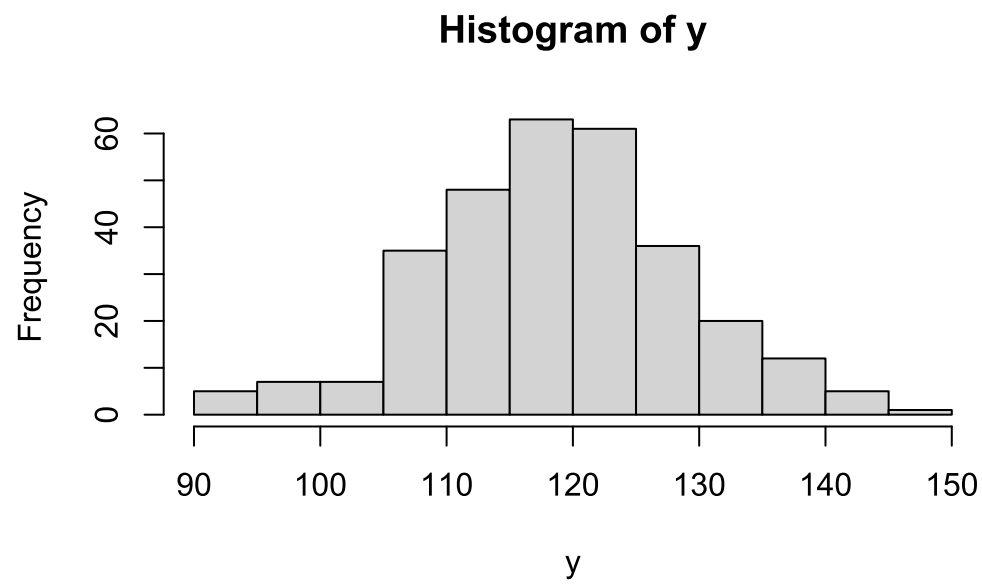
M5

It wouldn't with how I decided to standardize the variables, but in the spirit of the question, I would amend α to have a mean of 120 with an sd of 10. And let's say they grow an average of 2 ish centimeters per year. Giving $\log(\sigma)$ a mean of 0 and sd of 1 seems to give reasonable results from the prior predictive.

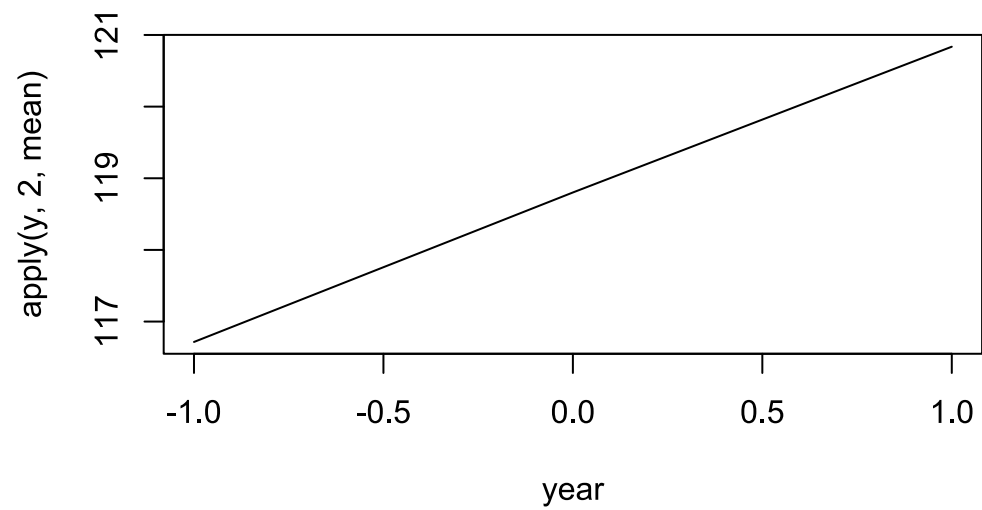
```
a <- rnorm(n, 120, 10)
b <- rnorm(n, 2, .5)
sigma <- rnorm(n, 0, 1)

y <- matrix(nrow = n, ncol = 3)
for(i in 1:3){
  y[,i] <- rnorm(n, a + b*(i-2), exp(sigma))
}

hist(y)
```



```
plot(apply(y, 2, mean) ~ year, type = "l")
```



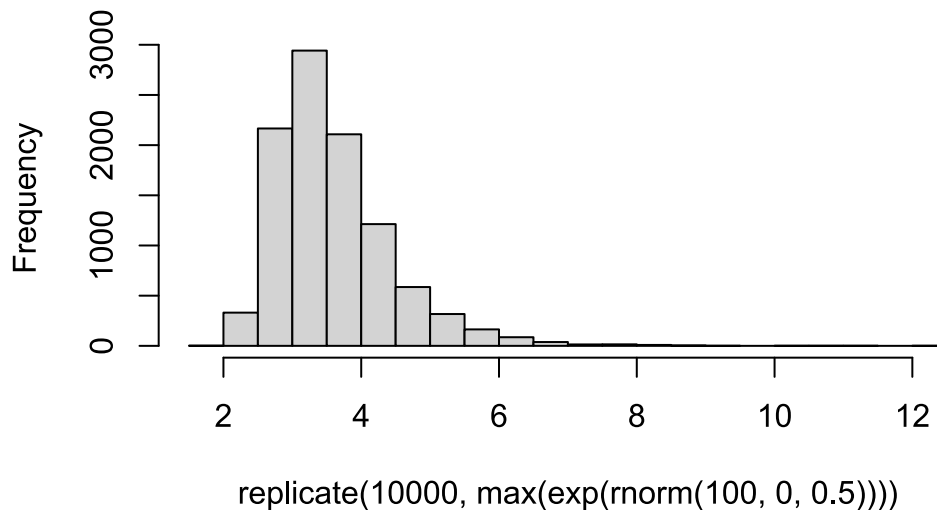
$$\begin{aligned}
height_i &\sim Normal(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta \times year_i \\
\alpha &\sim Normal(120, 10) \\
\beta &\sim Normal(2, .5) \\
\log(\sigma) &\sim Normal(0, 1)
\end{aligned}$$

M6

A variance of 64 is a standard deviation of 8, and my prior is predicting more variation than that, so I can probably tighten it down. Going to $\log(\sigma) \sim Normal(0, .5)$ seems to work well.

```
hist(replicate(1e4, max(exp(rnorm(100, 0, .5)))))
```

Histogram of replicate(10000, max(exp(rnorm(100, 0, 0.5))))



$$\begin{aligned}
height_i &\sim Normal(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta \times year_i \\
\alpha &\sim Normal(120, 10) \\
\beta &\sim Normal(2, .5) \\
\log(\sigma) &\sim Normal(0, .5)
\end{aligned}$$

Hard

H1

```

library(rethinking)
library(cmdstanr)
library(tidyverse)
data(Howell1)
d <- Howell1

# function for standardizing
stn <- function(x) (x - mean(x))/sd(x)

# data into a list for stan
dat <- list(
  N = nrow(d),
  weight = stn(d$weight),
  height = stn(d$height)
)

# compile the model
mod <- cmdstan_model("stan_models/height_mod.stan")

# fit the model to the data
fit_h1 <- mod$sample(
  data = dat,
  chains = 4,
  parallel_chains = 4,
  show_messages = F
)

# extract the parameters
alpha <- fit_h1$draws("a", format = "df")$a
beta <- fit_h1$draws("b", format = "df")$b
sigma <- fit_h1$draws("sigma", format = "df")$sigma

# make a vector of the observed weights and convert them to the standardized
scale
weight_obs <- c(46.95, 43.72, 64.78, 32.59, 54.63)
obs_stn <- (weight_obs - mean(d$weight))/sd(d$weight)

# posterior predictive distribution for the new observed weights
height_pred_stn <- sapply(obs_stn, function(x) rnorm(length(alpha), alpha + beta
* x, sigma))

# convert to un-standardized values and get into a dataframe
height_pred <- height_pred_stn * sd(d$height) + mean(d$height)
height_pred <- data.frame(height_pred)
colnames(height_pred) <- weight_obs

# print the values
height_pred %>%

```



```

pivot_longer(1:5, names_to = "weight",
             values_to = "height_pred") %>%
mutate(weight = as.numeric(weight)) %>%
group_by(weight) %>%
summarise(mu = mean(height_pred),
          lwr = HPDI(height_pred, .89)[1],
          upr = HPDI(height_pred, .89)[2]) %>%
round(2)

```

```

# A tibble: 5 × 4
  weight    mu   lwr   upr
  <dbl> <dbl> <dbl> <dbl>
1   32.6  133.  118.  147.
2   43.7  153.  138.  168.
3   47.0  158.  143.  172.
4   54.6  172.  157.  187.
5   64.8  190.  174.  204.

```

H2

```

# filtered data set so that all are lower than 18 years
d_young <- d %>%
  filter(age < 18)

# data list for stan
dat <- list(
  N = nrow(d_young),
  weight = stn(d_young$weight),
  height = stn(d_young$height)
)

# fit the model to the data
fit <- mod$sample(
  data = dat,
  chains = 4,
  parallel_chains = 4,
  show_messages = F
)

```

a

I'm gonna cheat and show 1 unit change (standardized) cause I don't want to do the work to convert at the moment.

```

# posterior mean of the slope
mean(fit$draws("b", format = "df")$b)

```

```
[1] 0.9424886
```

b

```
# extract the intercept and slope
a <- fit$draws("a", format = "df")$a
b <- fit$draws("b", format = "df")$b

# calculate the means and convert it back from standardized units
means <- sapply(stn(d_young$weight), function(x) a + b * x) * sd(d_young$height)
+ mean(d_young$height)

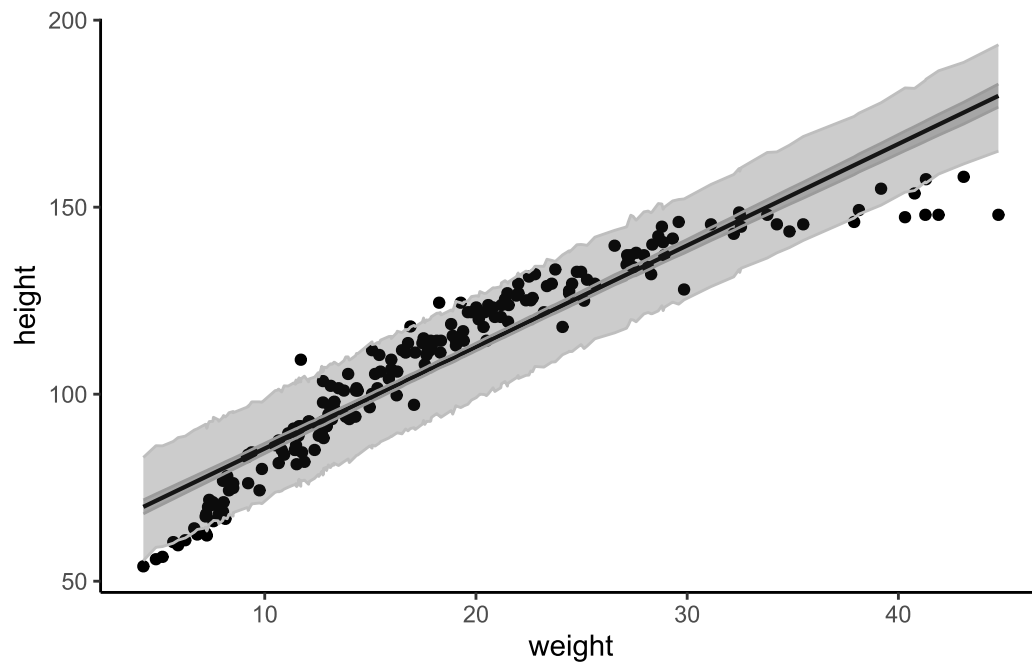
# get the mean of the means and the hpdi interval
mu_means <- apply(means, 2, mean)
upr_means <- apply(means, 2, function(x) HPDI(x, .89)[2])
lwr_means <- apply(means, 2, function(x) HPDI(x, .89)[1])

# extract the posterior predicted values
y_pred <- fit$draws("y_pred", format = "df")[,1:nrow(d_young)] *
sd(d_young$height) + mean(d_young$height)
```

Warning: Dropping 'draws_df' class as required metadata was removed.

```
# calculate the hpdi for the posterior predictive distribution
upr_pred <- apply(y_pred, 2, function(x) HPDI(x, .89)[2])
lwr_pred <- apply(y_pred, 2, function(x) HPDI(x, .89)[1])

# plot the data and the fit model
data.frame(weight = d_young$weight, height = d_young$height, mu_means,
upr_means,
           lwr_means, upr_pred, lwr_pred) %>%
  ggplot(aes(x = weight, y = height)) +
  geom_point() +
  geom_line(aes(x = weight, y = mu_means), linewidth = .75) +
  geom_ribbon(aes(x = weight, ymax = upr_means, ymin = lwr_means),
            color = "grey", alpha = .25) +
  geom_ribbon(aes(x = weight, ymax = upr_pred, ymin = lwr_pred),
            color = "grey", alpha = .25) +
  theme_classic()
```



c

This doesn't seem to be a very good fit, especially at low and high values of weight. Data aren't evenly distributed around the expectation. This might be because these data points are at the early stage of growth where the relationship between weight and height is changing. Maybe including a term for age would help.

H3

a

```
# data into a list for stan
dat <- list(
  N = nrow(d),
  log_weight = log(d$weight),
  height = d$height
)

# compile the model
mod <- cmdstan_model("stan_models/log_weight.stan")

# fit the model
fit <- mod$sample(
  data = dat,
  chains = 4,
  parallel_chains = 4,
  show_messages = F
)
```

```
# print the output
fit$print(c("a", "b", "sigma"))
```

variable	mean	median	sd	mad	q5	q95	rhat	ess_bulk	ess_tail
a	-23.71	-23.69	1.36	1.38	-25.94	-21.50	1.00	1155	1331
b	47.05	47.05	0.39	0.40	46.42	47.70	1.00	1159	1279
sigma	5.16	5.16	0.16	0.16	4.91	5.43	1.00	1447	1281

Since the predictor is on the log scale, and I don't think in log units, it is difficult to interpret the output. The intercept is about -24, i.e. when log of weight = 0, height would be negative, so there is no biological interpretation of the intercept. The slope is about 47, so one unit change in log weight corresponds to a 47 cm change in height.

b

```
# extract the intercept and slope from the fit
a <- fit$draws("a", format = "df")$a
b <- fit$draws("b", format = "df")$b

# calculate the predicted means and hpdi for the means
means <- sapply(log(d$weight), function(x) a + b * x)
mu_means <- apply(means, 2, mean)
upr_means <- apply(means, 2, function(x) HPDI(x)[2])
lwr_means <- apply(means, 2, function(x) HPDI(x)[1])

# extract the posterior predictive simulations
y_pred <- fit$draws("y_pred", format = "df")[,1:nrow(d)]
```

Warning: Dropping 'draws_df' class as required metadata was removed.

```
upr_pred <- apply(y_pred, 2, function(x) HPDI(x)[2])
lwr_pred <- apply(y_pred, 2, function(x) HPDI(x)[1])

# plot
data.frame(height = d$height, weight = d$weight, mu_means,
            upr_means, lwr_means, upr_pred, lwr_pred) %>%
  ggplot(aes(x = weight, y = height)) +
  geom_point(color = "lightblue") +
  geom_line(aes(x = weight, y = mu_means)) +
  geom_ribbon(aes(x = weight, ymax = upr_means, ymin = lwr_means),
             color = "grey", alpha = .75) +
  geom_ribbon(aes(x = weight, ymax = upr_pred, ymin = lwr_pred),
             color = "grey", alpha = .25) +
  theme_classic()
```

