Easy

E1

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

E2

There are two parameters: μ and σ.

E3

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

E4

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

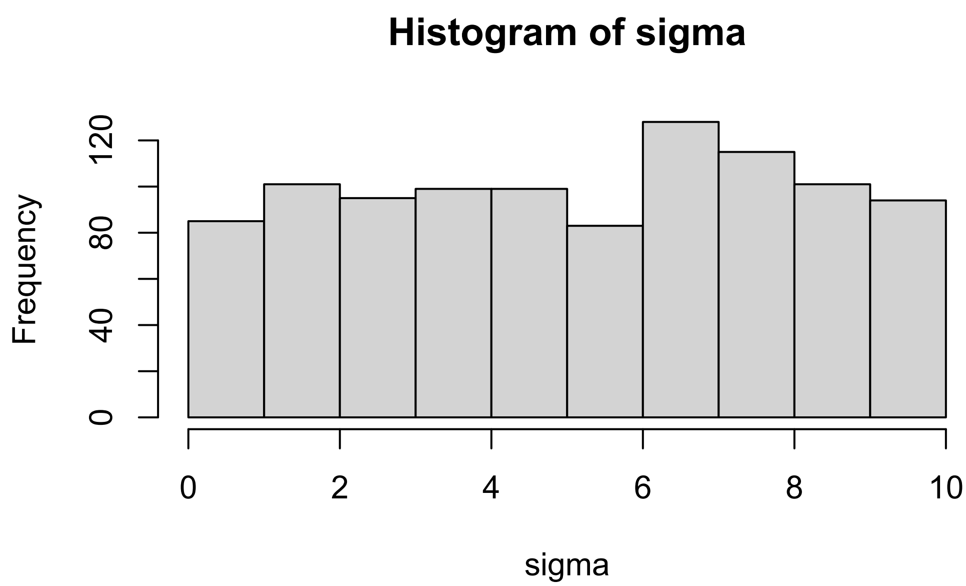
E5

There are 3 parameters: α, β, and σ.

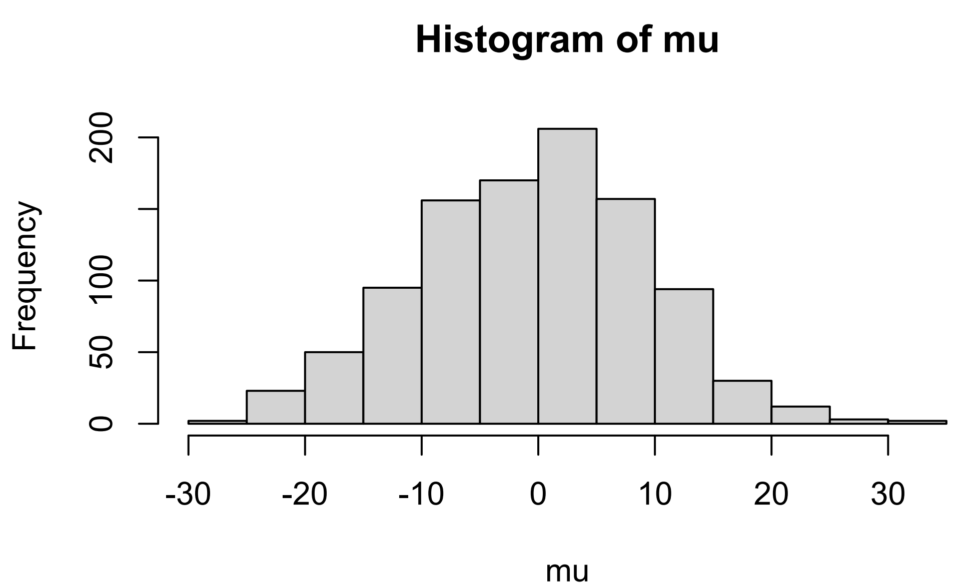
Medium

M1

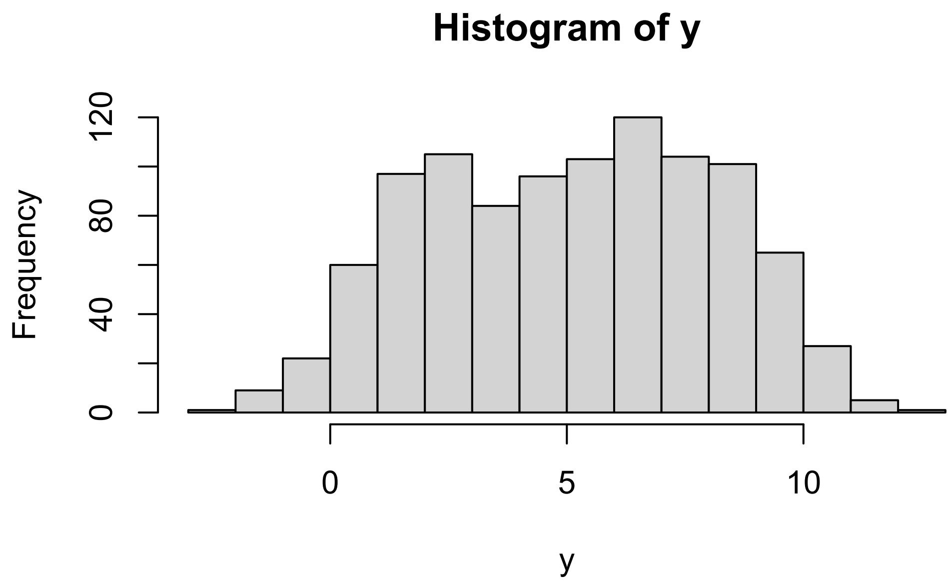
```r
sigma <- runif(1000, 0, 10)
mu <- rnorm(1000, 0, 10)
y <- rnorm(mu, sigma)
hist(sigma)
```




```
hist(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

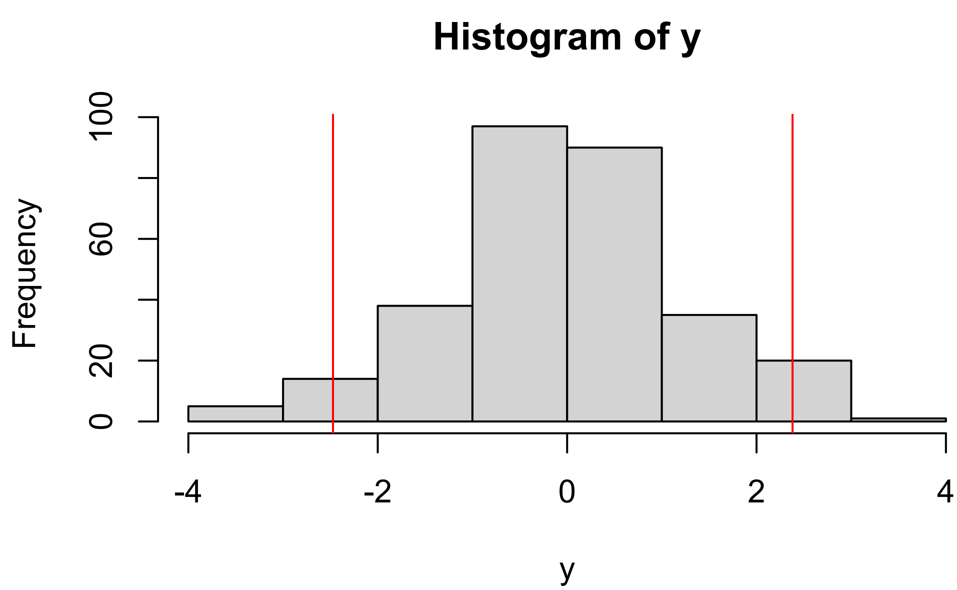
$$
\begin{aligned}
y_i &\sim Normal(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta x_i \\
\alpha &\sim Normal(0, 50) \\
\beta &\sim Uniform(0, 10) \\
\sigma &\sim Uniform(0, 50)
\end{aligned}
$$

M4

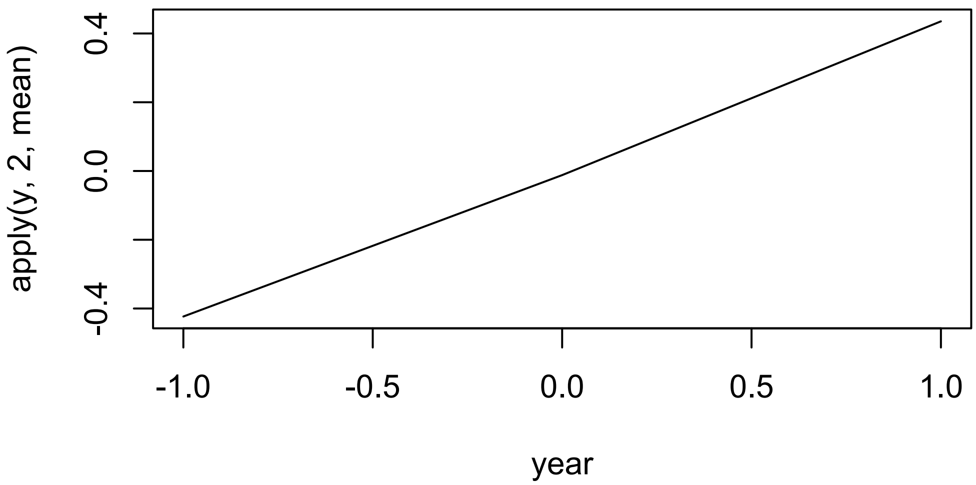
I will standardize height across all three years to make choosing priors easier. I will also code the first year as −1, the second as 0 and the third year as 1. This will make α very close to zero. Setting β to likely be positive (kids tend to grow). And $\log(\sigma)$ has a distribution around 1. These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



$$
\begin{aligned}
height_i &\sim Normal(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta \times year_i \\
\alpha &\sim Normal(0, .5) \\
\beta &\sim Normal(.5, .5) \\
\log(\sigma) &\sim Normal(0, .25)
\end{aligned}
$$

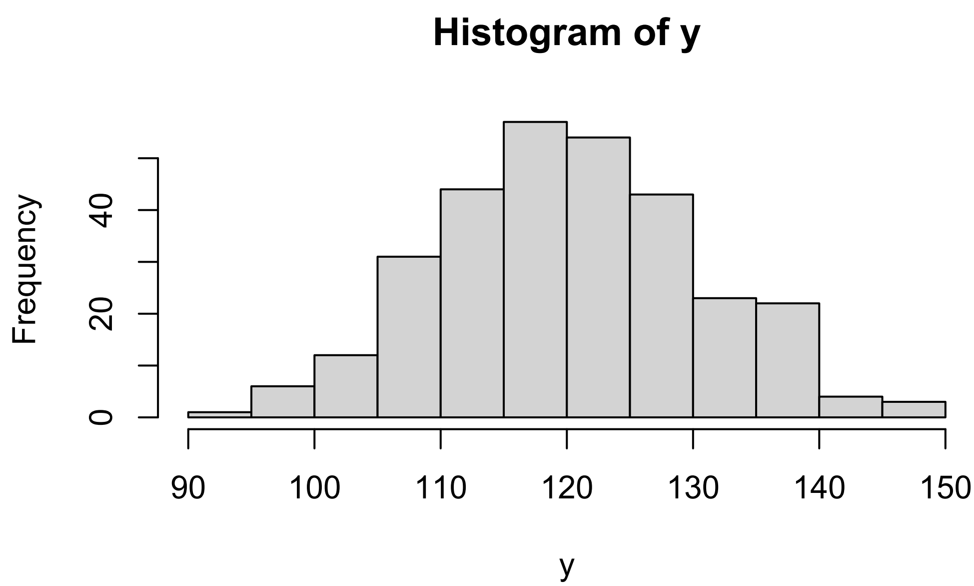
M5

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 0f 0 and sd of 1 seems to give reasonable results from the prior predictive.

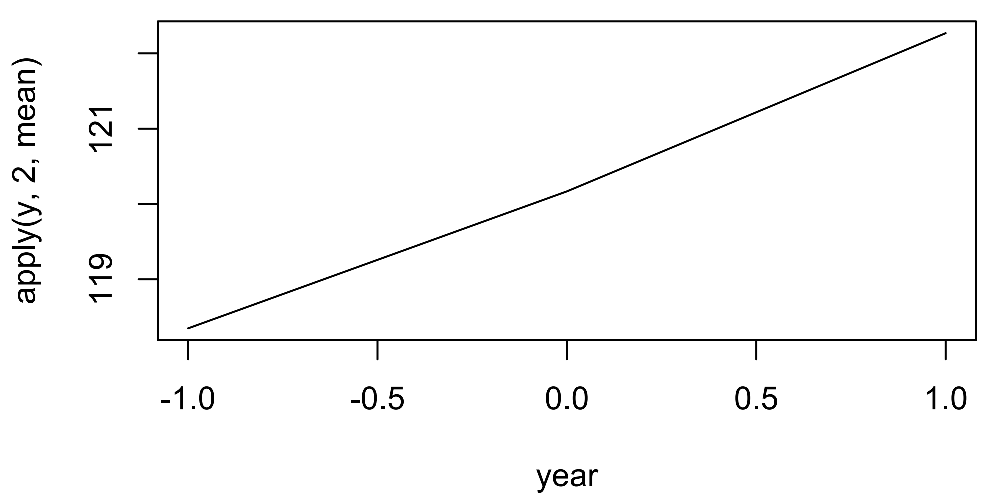
```
a <- rnorm(n,120, 10)
b <- rnorm(n,2,.5)
sigma <- rnorm(n,0, 1)

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")
```

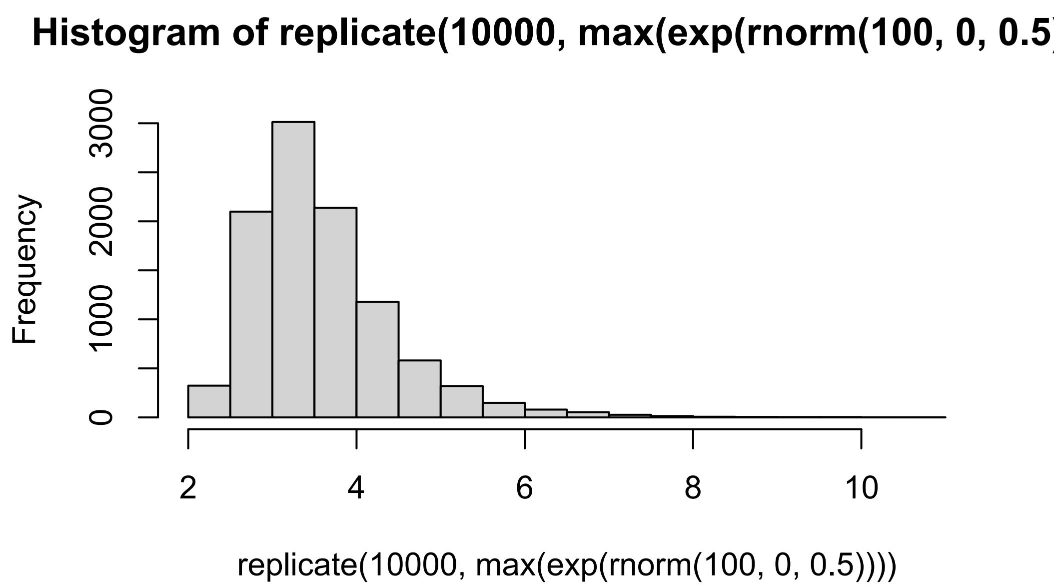


$$
\begin{aligned}
height_i &\sim Normal(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta \times year_i \\
\alpha &\sim Normal(120, 10) \\
\beta &\sim Normal(2, .5) \\
\log(\sigma) &\sim Normal(0, 1)
\end{aligned}
$$

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)))))
```



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

Hard

H1

```
```

```r
library(rethinking)
library(cmdstanr)
library(tidyverse)
data(Howell1)
d <- Howell1
stn <- function(x) (x - mean(x))/sd(x)

dat <- list(
  N = nrow(d),
  weight = stn(d$weight),
  height = stn(d$height)
)

mod <- cmdstan_model("stan_models/height_mod.stan")

fit_h1 <- mod$sample(
  data = dat,
  chains = 4,
  parallel_chains = 4,
  show_messages = F
)

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

weight_obs <- c(46.95, 43.72, 64.78, 32.59, 54.63)
obs_stn <- (weight_obs - mean(d$weight))/sd(d$weight)

height_pred_stn <- sapply(obs_stn, function(x) rnorm(length(alpha), alpha + beta
* x, sigma))

height_pred <- height_pred_stn * sd(d$height) + mean(d$height)
height_pred <- data.frame(height_pred)
colnames(height_pred) <- weight_obs

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.  117.  147.
2   43.7  153.  139.  168.
3   47.0  158.  144.  173.
4   54.6  172.  156.  187.
5   64.8  190.  175.  205.
```

H2

```
d_young <- d %>%
  filter(age < 18)

dat <- list(
  N = nrow(d_young),
  weight = stn(d_young$weight),
  height = stn(d_young$height)
)

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.

```
mean(fit$draws("b", format = "df")$b)
```

```
[1] 0.9417864
```

b

```
# means
a <- fit$draws("a", format = "df")$a
b <- fit$draws("b", format = "df")$b

means <- sapply(stn(d_young$weight), function(x) a + b * x) * sd(d_young$height)
+ mean(d_young$height)

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])

# predictid 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.
```

```
upr_pred <- apply(y_pred, 2, function(x) HPDI(x, .89)[2])
lwr_pred <- apply(y_pred, 2, function(x) HPDI(x, .89)[1])

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()
```