Fitting Bayesian Regression Models

Stat 340, Fall 2021

In this exericse we'll attempt to model the ridership among registered members of the Capital Bikeshare service in Washington, D.C. using the temperature (in °F). The data can be loaded using the below code:

```
data("bikes", package = "bayesrules")
```

Based on past bikeshare analyses, suppose we have the following prior understanding of this relationship:

- On an average temperature day, say 65 or 70 degrees for D.C., there are typically around 5000 riders, though this average could be somewhere between 3000 and 7000.
- For every one degree increase in temperature, ridership typically increases by 100 rides, though this average increase could be as low as 20 or as high as 180.
- At any given temperature, daily ridership will tend to vary with a moderate standard deviation of 1250 rides.

Tuning priors

(a) Tune a simple linear regression model to match our prior understanding. To translate the prior information for the intercept, it is easier to work with a *centered* version of temperature: $x_c = x - \overline{x}$. The intercept then tells us about the expected ridership on an average temperature day (i.e., when $x_c = 0$).

Use careful notation to write out the complete Bayesian structure of this model. (Hint: Consider using an exponential prior for σ.)

Solution: The prior information leads us to the following regression model:

$$\begin{aligned} Y_i | \beta_0, \beta_1, \sigma &\overset{\text{ind}}{\sim} \mu_\rangle, \sigma \\ \mu_i &= \beta_0 + \beta_1 x_{ci} \\ \beta_0 &\sim \mathcal{N}(5000, 1000) \\ \beta_1 &\sim \mathcal{N}(100, 40) \\ \sigma &\sim \text{Expo}(.0008) \end{aligned}$$

(b) To explore our combined prior understanding of the model parameters, simulate the Normal regression prior model for 1000 iterations.

You can simulate this directly in R using the `rnorm` and `rexp` functions.

```
S <- 1000
beta0 <- rnorm(S, 5000, 1000)
beta1 <- rnorm(S, 100, 40)
sigma <- rexp(S, .0008)
```

Note: You can also simulate from the prior model using JAGS by passing in `NA`s for the response variable.

```
prior_model <- "model{
  # Likelihood
  for(i in 1:n) {
y[i] ~ dnorm(mu[i], phi)
mu[i] <- beta0 + beta1 * x[i]
  }

  # Priors
  beta0 ~ dnorm(5000, pow(1000, -2))
  beta1 ~ dnorm(100, pow(40, -2))
  sigma ~ dexp(.0008)
  phi <- pow(sigma, -2)
}"
```

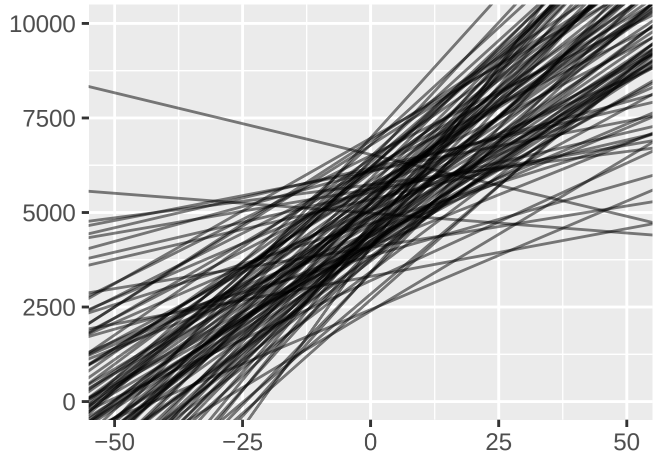
```
prior_data <- list(
  n = nrow(bikes),
  x = bikes$temp_actual - mean(bikes$temp_actual),
  y = rep(NA, nrow(bikes))
)

prior_sim <- run.jags(
  prior_model,
  data = prior_data,
  n.chains = 1,
  sample = 1000,
  monitor = c("beta0", "beta1", "sigma", "y"),
  silent.jags = TRUE
)
```

(c) Plot 100 prior plausible model lines ($\beta_0 + \beta_1 x_c$) and four data sets simulated under the priors. Since you used a centered version of temperature, you can either stick with this centered predictor, or "back transform" the plausible lines. To back transform, notice that $\beta_0 + \beta_1 x_c = (\beta_0 - \beta_1 \overline{x}) + \beta_1 x$.

Describe our overall prior understanding of the relationship between ridership and humidity.

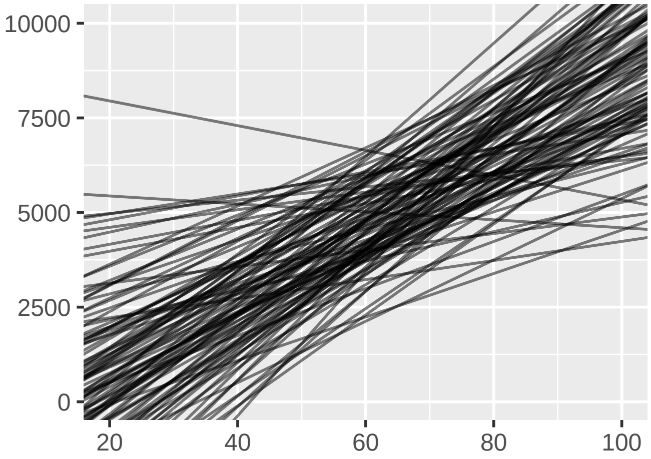
First, let's plot the 100 prior plausible model lines with the centered predictors. One way to do this is using `geom_abline` in `ggplot2`:

```
library(tidyverse)
prior_lines <- data.frame(beta0 = beta0, beta1 = beta1) %>%
  mutate(group = row_number())

ggplot(prior_lines[1:100,]) +
  geom_abline(aes(intercept = beta0, slope = beta1, group = group), alpha = 0.5) +
  lims(x = c(-50, 50),
   y = c(20, 10000))
```



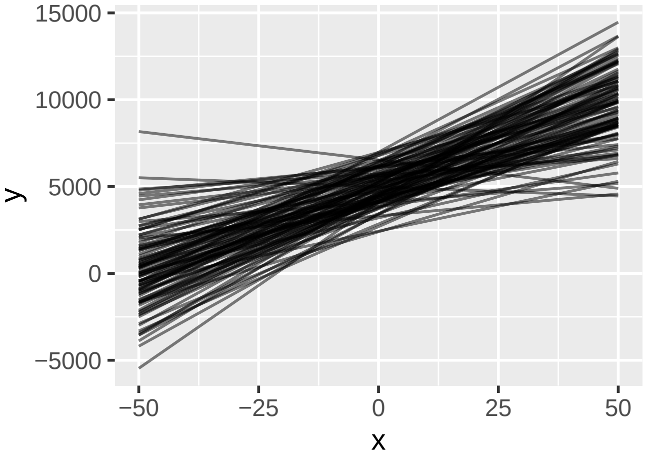
To create an uncentered version on the original x-scale, we need to alter `beta0`:

```
xbar <- mean(bikes$temp_actual)
ggplot(prior_lines[1:100,]) +
  geom_abline(aes(intercept = beta0 - beta1 * xbar, slope = beta1, group = group), alpha = 0.5) +
  lims(x = c(20, 100),
   y = c(20, 10000))
```



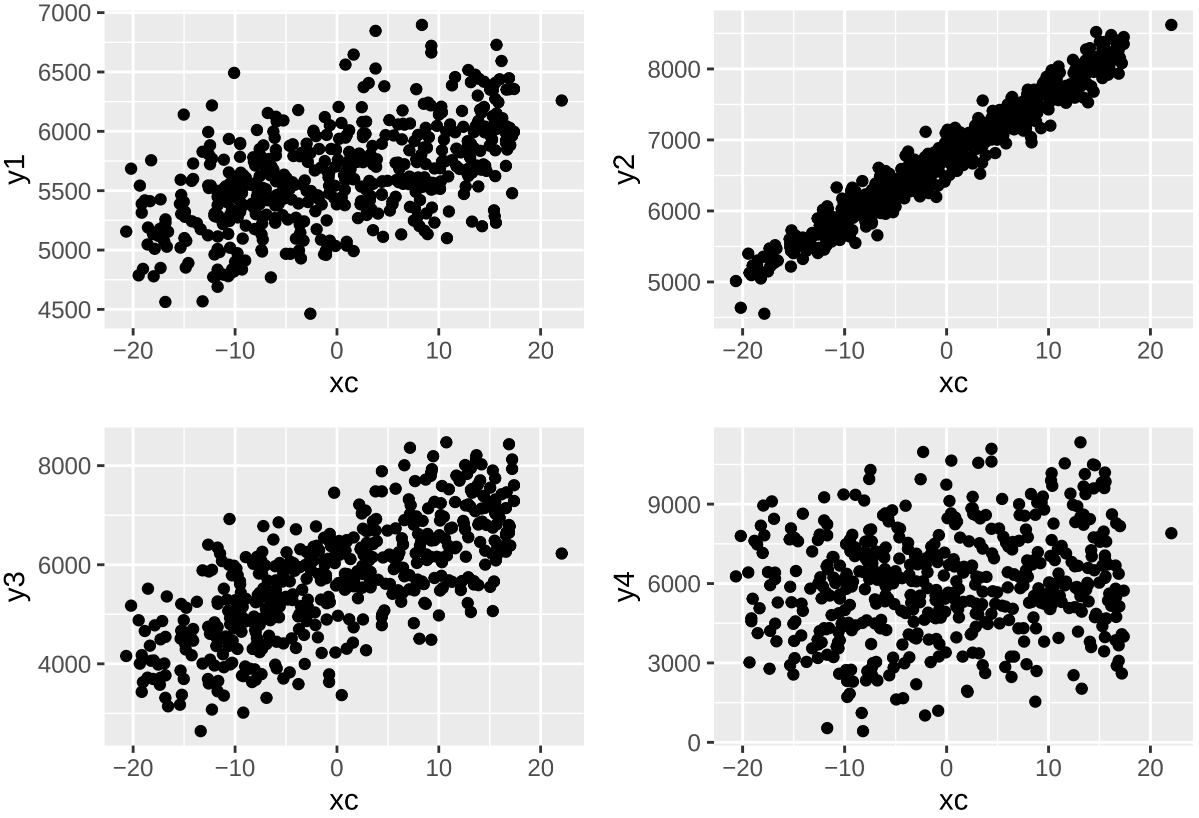
An alternative way to do this is to fit a line through two points on the edge of plausible x_c values using `geom_segment()`. Let's assume that centered temperature ranges from -50 to 50

```
prior_lines <- data.frame(beta0 = beta0, beta1 = beta1, x = -50, xend = 50) %>%
  mutate(y = beta0 + beta1 * x, yend = beta0 + beta1 * xend)

ggplot(prior_lines[1:100,]) +
  geom_segment(aes(x = x, y = y, xend = xend, yend = yend), alpha = 0.5)
```



Now, let's simulate four data sets under the priors:

```
xc <- bikes$temp_actual - mean(bikes$temp_actual)
y1 <- rnorm(500, mean = beta0[1] + beta1[1] * xc, sd = sigma[1])
y2 <- rnorm(500, mean = beta0[2] + beta1[2] * xc, sd = sigma[2])
y3 <- rnorm(500, mean = beta0[3] + beta1[3] * xc, sd = sigma[3])
y4 <- rnorm(500, mean = beta0[4] + beta1[4] * xc, sd = sigma[4])

library(patchwork)
p1 <- ggplot(NULL) + geom_point(aes(x = xc, y = y1))
p2 <- ggplot(NULL) + geom_point(aes(x = xc, y = y2))
p3 <- ggplot(NULL) + geom_point(aes(x = xc, y = y3))
p4 <- ggplot(NULL) + geom_point(aes(x = xc, y = y4))

p1 + p2 + p3 + p4 + plot_layout(nrow = 2)
```



Regardless of the plot, your discussion should center around whether the plot reflects your the prior understanding you were attempting to model.

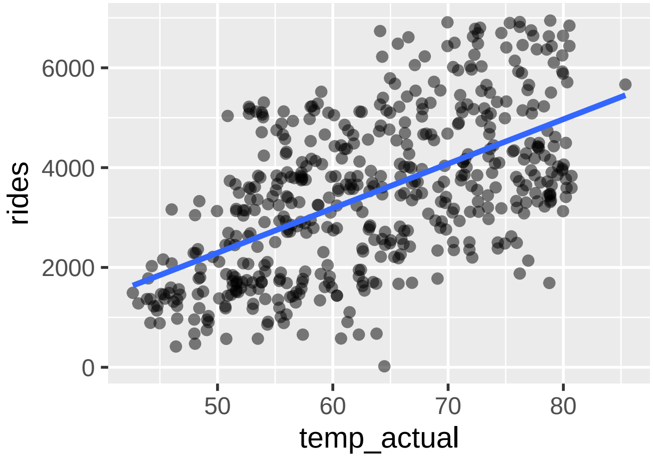
EDA

With the priors in place, it's time to examine the data.

(a) Plot and discuss the observed relationship between ridership and humidity in the bikes data.

From the scatterplot, we see a moderate, positive, linear association between the temperature and number os riders.

```r
ggplot(bikes, aes(x = temp_actual, y = rides)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE)
```



(b) Does simple Normal regression seem to be a reasonable approach to modeling this relationship? Explain.

Yes, there appears to be a postive linear association between temperature and ridership.

Model fitting

We can now simulate our posterior model of the relationship between ridership and temperature, a balance between our prior understanding and the data.

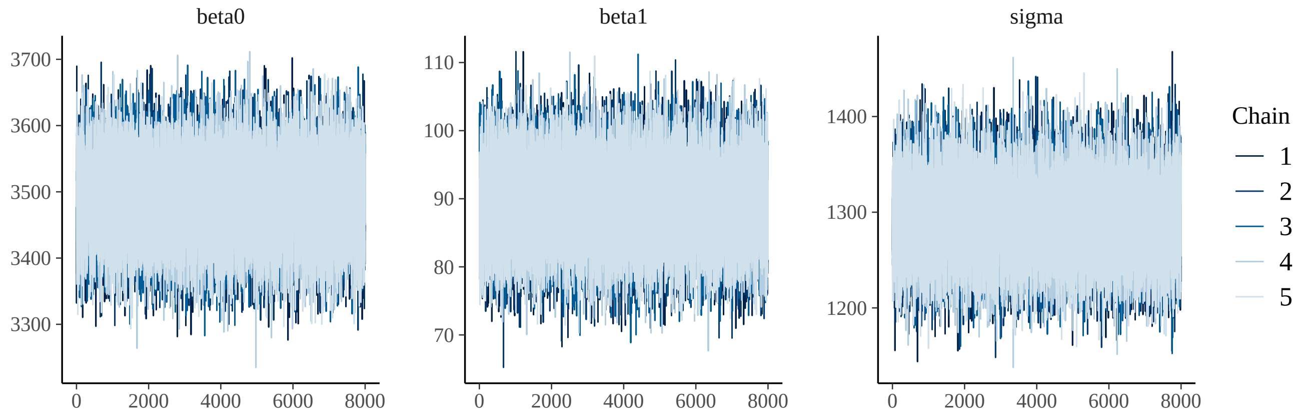
(a) Use JAGS to simulate the Normal regression posterior model. Do so with 5 chains run for 8000 iterations each.

```
post_model <- "model{
  # Likelihood
  for(i in 1:n) {
y[i] ~ dnorm(mu[i], phi)
mu[i] <- beta0 + beta1 * x[i]
  }

  # Priors
  beta0 ~ dnorm(5000, pow(1000, -2))
  beta1 ~ dnorm(100, pow(40, -2))
  sigma ~ dexp(.0008)
  phi <- pow(sigma, -2)
}"

post_data <- list(
  n = nrow(bikes),
  x = bikes$temp_actual - mean(bikes$temp_actual),
  y = bikes$rides
)

posterior <- run.jags(
  post_model,
  data = post_data,
  n.chains = 5,
  adapt = 1000,
  burnin = 5000,
  sample = 8000,
  monitor = c("beta0", "beta1", "sigma")
)
```

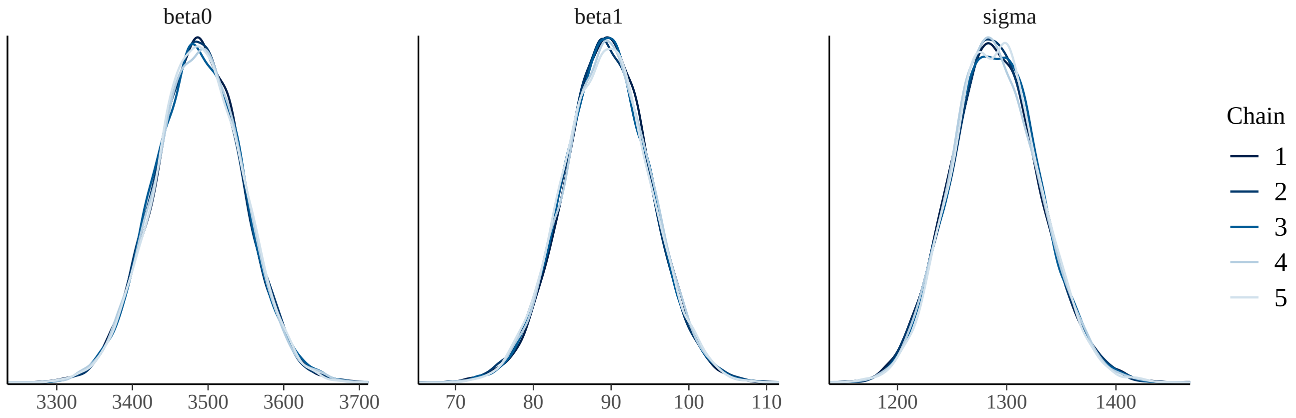
(b) Perform and discuss some MCMC diagnostics to determine whether or not we can "trust" these simulation results.

All of our diagnostics check out. The chains appear to be stationary and well mixed. Further, we see no issues with autocorrelation.

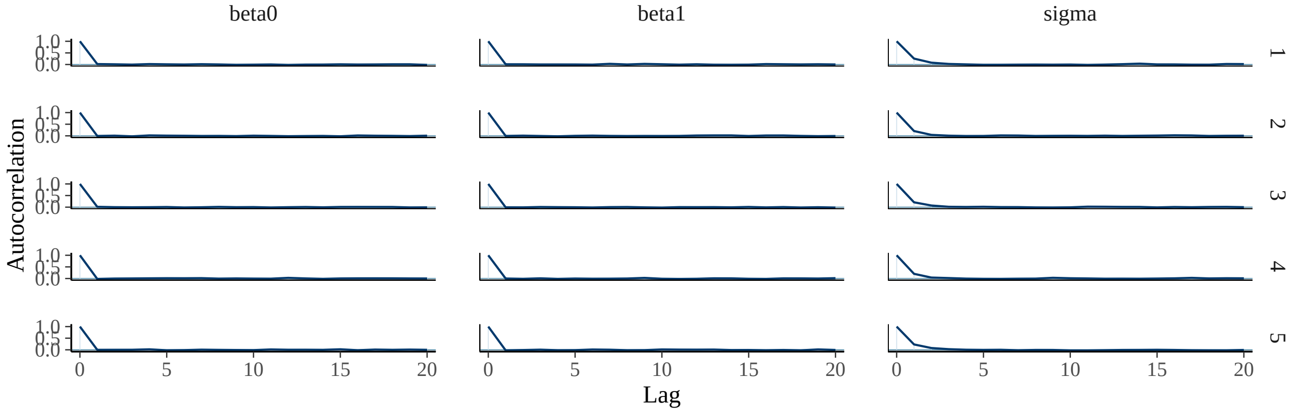
```
mcmc_trace(posterior$mcmc)
```



```
mcmc_dens_overlay(posterior$mcmc)
```

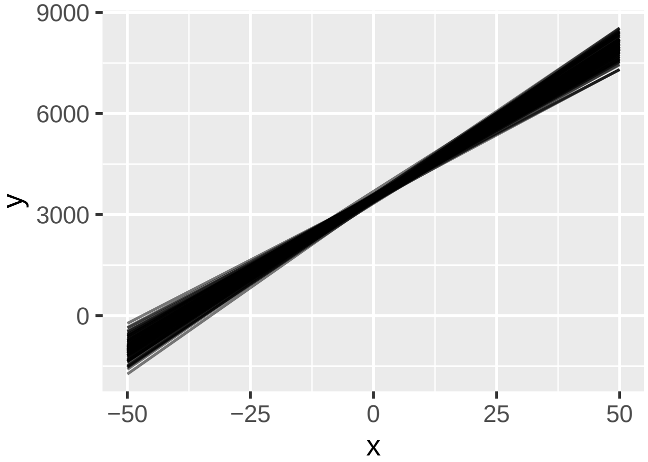


```
mcmc_acf(posterior$mcmc)
```



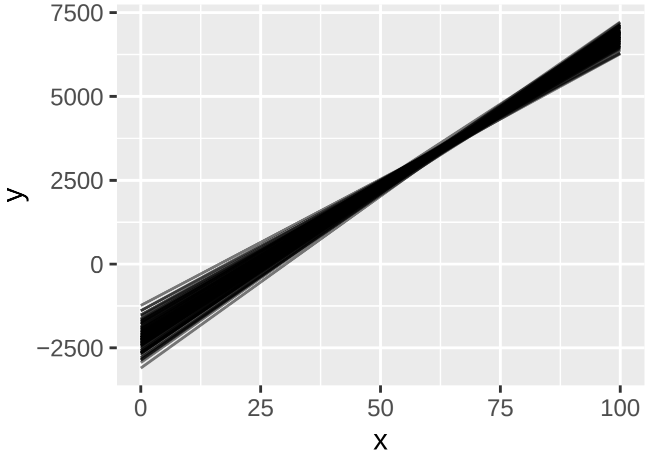
(c) Plot 100 posterior model lines for the relationship between ridership and temperature. Compare and contrast these to the prior model lines from above.

```
posterior_lines <- posterior$mcmc[[1]][1:100,c("beta0", "beta1")] %>%
  as.data.frame() %>%
  mutate(x = -50, xend = 50, y = beta0 + beta1 * x, yend = beta0 + beta1 * xend)

ggplot(posterior_lines) +
  geom_segment(aes(x = x, y = y, xend = xend, yend = yend), alpha = 0.5)
```



We can also put this on the original x-axis

```
posterior_lines2 <- posterior$mcmc[[1]][1:100,c("beta0", "beta1")] %>%
  as.data.frame() %>%
  mutate(x = 0, xend = 100, y = beta0 - beta1 * xbar + beta1 * x, yend = beta0 - beta1 * xbar + beta1
ggplot(posterior_lines2) +
  geom_segment(aes(x = x, y = y, xend = xend, yend = yend), alpha = 0.5)
```



Posterior inference

Finally, let's dig deeper into our posterior understanding of the relationship between ridership and humidity.

(a) Provide a summary of your posterior model, including 95% credible intervals.

We can easily `print` this type of summary.

```
print(posterior, digits = 3)
```

```
##
## JAGS model summary statistics from 40000 samples (chains = 5; adapt+burnin = 6000):
##
##       Lower95 Median Upper95 Mean   SD Mode  MCerr MC%ofSD SSeff    AC.10 psrf
## beta0    3373   3487    3600 3487   58 3486   0.29     0.5 40018  -0.0032    1
## beta1    78.7   89.5     101 89.5  5.6 89.3 0.0278     0.5 40376 -0.00246    1
## sigma    1213   1289    1372 1290 40.6 1287  0.257     0.6 25048 -0.00455    1
##
## Total time taken: 10 seconds
```

(b) Interpret the posterior median value of the σ parameter.

The posterior median of σ is approximately 1289.029.

On average, we can expect the observed ridership on a given day to fall about 1289 rides from the average ridership on days of the same temperature.

Note: If you want to pool the draws from all of the chains and calculate this (or CIs) manually, then the `tidy_draws()` function from the `tidybayes` R package is worth exploring.

(c) Interpret the 95% posterior credible interval for the temperature coefficient, β_1.

From our summary, we see the 95% posterior credible interval for the temperature coefficient is: 78.72 to 100.69

For every one degree increase in temperature, we expect ridership to increase between roughly 78.72 and 100.69 riders, with 0.95 posterior probability.

(d) Do we have ample posterior evidence that there's a positive association between ridership and temperature? Explain.

Yes, 0 is far below our 95% credible interval.

Prediction

Tomorrow is supposed to be 67°F in Washington, D.C. What levels of ridership should we expect?

(a) Simulate two posterior models: the posterior model for the typical number of riders on 67°F days; and the posterior predictive model for the number of riders tomorrow.

First, let's simulate from the model for the typical (average) number of riders. To do this, we generate $\mu^{(1)}, \dots, \mu^{(S)}$ draws by calculating $\mu^{(i)} = \beta_0^{(i)} + \beta_1^{(i)} x_i$. Here, it's important to know whether you need to center $x = 67$ or not. Since I used centered variables in my `posterior` model, I first need to center 67.

```
library(tidybayes)

mean67 <- tidy_draws(posterior) %>%
  mutate(
xc = 67 - mean(bikes$temp_actual),
mu = beta0 + beta1 * xc
  )
```

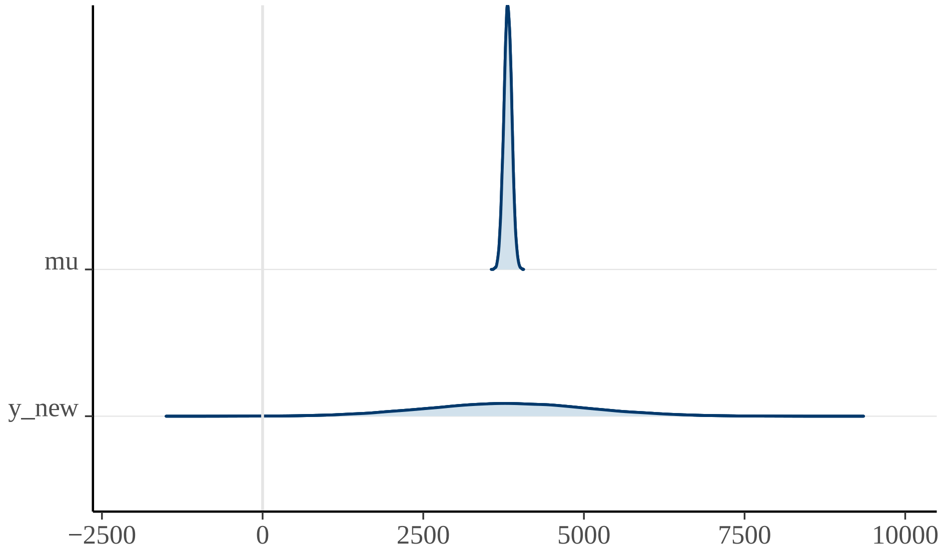
To simulate from the posterior predictive distribution we need to generate a value of our response for each $\mu^{(i)}$, so we draw $y_{new}^{(i)} \sim \mathcal{N}(\mu^{(i)}, \sigma^{(i)})$.

```
predict67 <- tidy_draws(posterior) %>%
  mutate(
xc = 67 - mean(bikes$temp_actual),
mu = beta0 + beta1 * xc,
y_new = rnorm(length(mu), mu, sd = sigma)
  )
```

(b) Construct, discuss, and compare density plot visualizations for the two separate posterior models in the previous part.

You could use `mcmc_areas_ridges` in `bayesplot`, or you could use `ggplot`. Either way, you should notice that the distributions are centered at the same location, but that the variability is far larger for the new observation.

```
mcmc_areas_ridges(predict67, pars = c("mu", "y_new"))
```



(c) Calculate and interpret an 80% posterior prediction interval for the number of riders tomorrow

We expect the number of riders tomorrow, a 67°F day, to be between approximately 2163 and 5472 riders, with 80% posterior probability.

```
quantile(predict67$y_new, probs = c(0.1, 0.9))
```

```
##      10%      90% 
## 2163.355 5471.676
```