

Bayesian Linear Regression

PSY517 Quantitative Analysis III

Derek Powell

Module 2

2021-09-07

Workflow

- propose model structure
- prior predictive checks
- fit model
- posterior predictive checks
- prediction, inference, etc.

- propose model structure
- prior predictive checks
- fit model
- posterior predictive checks
- prediction, inference, etc.

Proposing model structure

2021-09-07

Bayesian Linear Regression
└ Proposing model structure

Proposing model structure

Capital Bikeshare is metro DC's bikeshare service, with 4,500 bikes and 500+ stations across 7 jurisdictions: Washington, DC.; Arlington, VA; Alexandria, VA; Montgomery, MD; Prince George's County, MD; Fairfax County, VA; and the City of Falls Church, VA. Designed for quick trips with convenience in mind, it's a fun and affordable way to get around.

We will consider a dataset that includes the total number of Capital Bikeshare rides from 500 separate days.

2021-09-07

Bayesian Linear Regression
└ Proposing model structure

└ Capital Bikeshare

Capital Bikeshare

Capital Bikeshare is metro DC's bikeshare service, with 4,500 bikes and 500+ stations across 7 jurisdictions: Washington, DC.; Arlington, VA; Alexandria, VA; Montgomery, MD; Prince George's County, MD; Fairfax County, VA; and the City of Falls Church, VA. Designed for quick trips with convenience in mind, it's a fun and affordable way to get around.

We will consider a dataset that includes the total number of Capital Bikeshare rides from 500 separate days.

- We are interested in relationship between the weather, specifically temperature, and rides
- We think rides is a function of temperature. Roughly:

$$rides = f(temperature)$$

- We are interested in relationship between the weather, specifically temperature, and rides
- We think rides is a *function* of temperature. Roughly:

$$rides = f(temperature)$$

Proposing a linear model

- We will assume f is a linear function

$$rides = \alpha + \beta * temperature$$

- Let's draw what this might look like
- Will this assumption work?

Bayesian Linear Regression

└ Proposing model structure

└ Proposing a linear model

2021-09-07

draw on board:

- show impossible predictions (e.g. negative rides)
- willingness to predict for impossible values

Proposing a linear model

- We will assume f is a linear function

$$rides = \alpha + \beta * temperature$$

- Let's draw what this might look like
- Will this assumption work?

- Can do a good job describing data generated by many different processes
- Not always (or often) good models of the true domain process
- Should not be taken too literally
 - They will happily make impossible predictions

2021-09-07

Bayesian Linear Regression

└ Proposing model structure

└ Linear regression as geocentric model

- Linear regression as geocentric model
- Can do a good job describing data generated by many different processes
 - Not always (or often) good models of the true domain process
 - Should not be taken too literally
 - They will happily make impossible predictions

Is a linear model appropriate?

- *temperature* takes on a limited range of values
- *rides* is bounded at zero in principle
 - But we know there are about 5000 rides per day on average, so may not be near zero

Bayesian Linear Regression

└ Proposing model structure

└ Is a linear model appropriate?

So it seems reasonable rides will not often be near zero.

If we ask our linear golem to make predictions about temperatures from neptune or venus (or wisconsin or phoenix) we could be in trouble. But as long as we don't blindly trust our golem, it could give a useful description and predictions

- *temperature* takes on a limited range of values
- *rides* is bounded at zero in principle
 - But we know there are about 5000 rides per day on average, so may not be near zero

Let's begin with the **likelihood** of our model:

$$y_i \stackrel{iid}{\sim} \text{Normal}(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta x_i$$

In English this says:

- Each observations y_i will be distributed normally with mean μ_i and standard deviation σ .
- Where μ_i is a linear function of the x_i predictor values associated with each observation.

2021-09-07

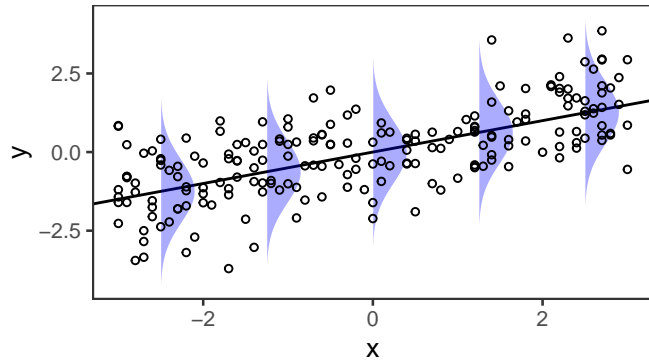
$$y_i \stackrel{iid}{\sim} \text{Normal}(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta x_i$$

- Each observations y_i will be distributed normally with mean μ_i and standard deviation σ .
- Where μ_i is a linear function of the x_i predictor values associated with each observation.

Visualizing linear regression with a normal likelihood

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

$$\mu_i = 0 + .5x_i, \sigma = 1$$

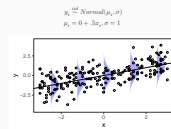


Bayesian Linear Regression

└ Proposing model structure

└ Visualizing linear regression with a normal likelihood

Visualizing linear regression with a normal likelihood



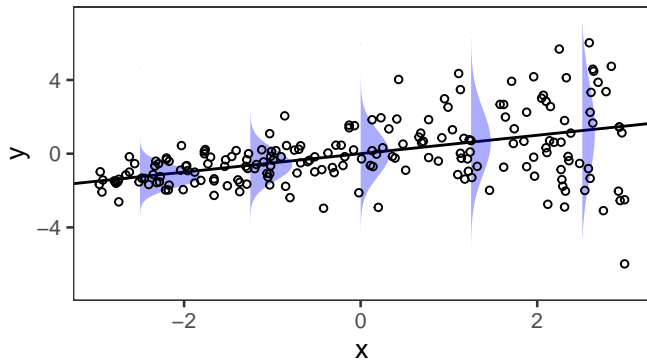
2021-09-07

Extension: Heteroskedasticity

$$y_i \stackrel{iid}{\sim} \text{Normal}(\mu_i, \sigma_i)$$

$$\mu_i = 0 + .5x$$

$$\log(\sigma_i) = .2 + .3x$$



Bayesian Linear Regression

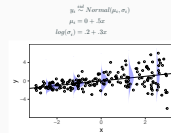
└ Proposing model structure

└ Extension: Heteroskedasticity

2021-09-07

- we'll work with some models like that later in the course

Extension: Heteroskedasticity



Is a normal likelihood appropriate for y ?

- Capital Bike Share's members are able to take bikes for a ride for a fee
- The number of rides per day is the cumulative sum of all of the individual members' decisions to ride
- This is the kind of process that is likely to converge to an approximately normal distribution, via the *central limit theorem*.

Bayesian Linear Regression

└ Proposing model structure

└ Is a normal likelihood appropriate for y ?

- Capital Bike Share's members are able to take bikes for a ride for a fee
- The number of rides per day is the cumulative sum of all of the individual members' decisions to ride
- This is the kind of process that is likely to converge to an approximately normal distribution, via the *central limit theorem*.

2021-09-07

Let's demonstrate to ourselves that a normal distribution could be appropriate for these data.

```
N <- 1000
trips <- 0:8
trip_probs <- c(.2, .05, .35, .15, .1, .05, .05, .03, .02)
ride_probs <- .5
```

```
person_trips <- sample(trips, N, replace=TRUE, prob=trip_probs)
person_rides <- rbinom(N, person_trips, ride_probs)
```

```
person_trips[1:20]
```

```
## [1] 0 3 3 0 2 3 2 3 0 3 3 0 2 0 5 4 3 0 2 6
```

```
person_rides[1:20]
```

```
## [1] 0 0 1 0 1 3 2 1 0 2 2 0 1 0 4 3 1 0 2 2
```

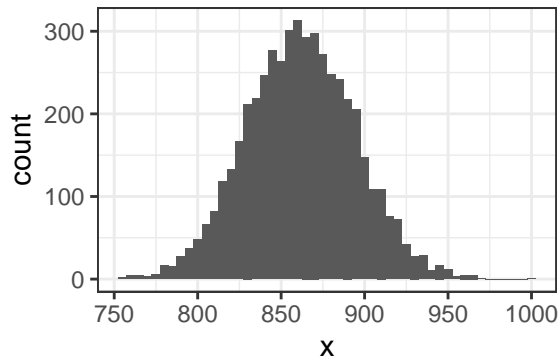
- it's a membership model, so we have some number of members.
- each of them may have some number of trips they need to make during the day, let's say between 0 and 8
- they each have a probability of choosing to bike on each trip
- the probability they each choose to bike is a function of the weather; but we'll fix that here



Simulating riders: showing y is plausibly normal

```
sim_day <- function(){  
  person_trips <- sample(trips, N, replace=TRUE, prob=trip_probs)  
  person_rides <- rbinom(N, person_trips, ride_probs)  
  sum(person_rides)  
}
```

```
gghist(replicate(5000, sim_day()), binwidth=5)
```

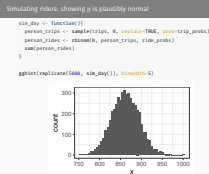


Bayesian Linear Regression

Proposing model structure

Simulating riders: showing y is plausibly normal

- This is a plausible story, but we can't evaluate it here since all we have measured is total rides. This is what McElreath means by saying gaussian distributions cannot reliably identify "microprocesses"



2021-09-07

Priors and prior predictive checks

2021-09-07

Bayesian Linear Regression
└ Priors and prior predictive checks

Priors and prior predictive checks

Priors

For our model:

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

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

We will use need priors for α , β , and σ

For our model:

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

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

We will we need priors for α , β , and σ

- Uninformative priors
- Weakly informative priors
- Informative priors

But there are no such thing as “correct” or “true” priors

- there is no “correct” prior
- these approaches lie on a continuum
- in this course we will focus on the latter two

• Uninformative priors
• Weakly informative priors
• Informative priors
But there are no such thing as “correct” or “true” priors

- For linear regression: priors defined on standardized variables
- Restricts to “reasonable” range of parameter values, but without incorporating any specific information
- One possible example would be:

$$\alpha \sim \text{Normal}(0, 3)$$

$$\beta \sim \text{Normal}(0, 3)$$

$$\sigma \sim \text{HalfCauchy}(5)$$

we'll see what this looks like later

- For linear regression: priors defined on standardized variables
- Restricts to “reasonable” range of parameter values, but without incorporating any specific information
- One possible example would be:

$$\begin{aligned}\alpha &\sim \text{Normal}(0, 3) \\ \beta &\sim \text{Normal}(0, 3) \\ \sigma &\sim \text{HalfCauchy}(5)\end{aligned}$$

What domain knowledge do we have or can we get?

- Start knowing that the average number of rides per day is about 5000
- Will cooler temperatures lead more people to ride or fewer?
 - What range of temperatures are typical in D.C.?

- informative priors based on our knowledge BEFORE we see the data
- In PHX higher temperatures would probably mean fewer riders. In D.C. it's less clear. It does get pretty cold, but average Jan highs are 40 degrees, which might be more pleasant for a ride than 90 degrees.

I'll start by proposing the following priors based on what I know so far about these data

$$\alpha \sim \text{Normal}(5000, 2000)$$

$$\beta \sim \text{Normal}(150, 300)$$

$$\sigma \sim \text{Exponential}(1/250)$$

We'll evaluate and possibly improve these priors in a moment.

2021-09-07

Proposing informative priors

- I know α should be about 5000 but I'm not sure how confident I can be
- I think β will be positive but it could be negative too, and I don't know how large a number it will be
- I'm not sure what σ will be, but the expectation of an exponential is the inverse of its rate parameter (so 250 here)

$$\begin{aligned}\alpha &\sim \text{Normal}(5000, 2000) \\ \beta &\sim \text{Normal}(150, 300) \\ \sigma &\sim \text{Exponential}(1/250)\end{aligned}$$

Bayesian regression in **brms**

2021-09-07

Bayesian Linear Regression
└ Bayesian regression in **brms**

Bayesian regression in **brms**

We will use the **brms** package to fit all of the Bayesian regression models we will learn about in this class.

Why **brms** ?

- **brms** supports nearly any model you could ever wish, all within a single common syntax and with many additional helpful functions.

Why not **brms**?

- Fitting models with **brms** is slower than with functions like `lm()`

2021-09-07

Bayesian Linear Regression

└ Bayesian regression in **brms**

└ The **brms** package

The **brms** package

We will use the **brms** package to fit all of the Bayesian regression models we will learn about in this class.

Why **brms** ?

- **brms** supports nearly any model you could ever wish, all within a single common syntax and with many additional helpful functions.

Why not **brms**?

- Fitting models with **brms** is slower than with functions like `lm()`

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

- α represents the expected value of μ_i when the temperature is 0, but D.C. never really reaches 0° Fahrenheit
- Centering predictors x_i as $x_i - \bar{x}$ let's α represents the number of rides on an average temperature day, rather than a 0° day.
- Conveniently, **brms** does this centering for us internally, and will interpret a prior set on the intercept term relative to the centered predictors

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

- α represents the expected value of μ_i when the temperature is 0, but D.C. never really reaches 0° Fahrenheit
- Centering predictors x_i as $x_i - \bar{x}$ let's α represents the number of rides on an average temperature day, rather than a 0° day.
- Conveniently, **brms** does this centering for us internally, and will interpret a prior set on the intercept term relative to the centered predictors

Recall our initial proposal for priors

$$\alpha \sim \text{Normal}(5000, 2000)$$

$$\beta \sim \text{Normal}(150, 300)$$

$$\sigma \sim \text{Exponential}(1/250)$$

Let's evaluate these priors.

Some candidate informative priors

- I know α should be about 5000 but I'm not sure how confident I can be
- I think β will be positive but it could be negative too, and I don't know how large a number it will be
- I'm not sure what σ will be, but the expectation of an exponential is the inverse of its rate parameter (so 250 here)

Recall our initial proposal for priors

```

$$\alpha \sim \text{Normal}(5000, 2000)$$

$$\beta \sim \text{Normal}(150, 300)$$

$$\sigma \sim \text{Exponential}(1/250)$$

```

Let's evaluate these priors.

2021-09-07

Loading the bikes data

We haven't even loaded the data yet because priors need to come "prior" to the data: Your priors should be based on general domain knowledge and **not** on the data you are analyzing.

But, to set up our model we will need to load the data.

```
bikes <- read_csv("../data/bikes.csv")
```

Bayesian Linear Regression

Bayesian regression in brms

Loading the bikes data

- With informative priors must come before data
- But with weakly informative it could be ok to look at the data to get a sense of the scale that the data is on

2021-09-07

Loading the bikes data

We haven't even loaded the data yet because priors need to come "prior" to the data. Your priors should be based on general domain knowledge and not on the data you are analyzing.

But, to set up our model we will need to load the data.

```
bikes <- read_csv("../data/bikes.csv")
```

Let's translate our model into **brms** syntax:

Model

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

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

$$\alpha \sim \text{Normal}(5000, 2000)$$

$$\beta \sim \text{Normal}(150, 300)$$

$$\sigma \sim \text{Exponential}(1/250)$$

brms syntax

```
fit <- brm(
  rides ~ temp_feel,
  prior =
    prior(normal(5000, 2000), class="Intercept") +
    prior(normal(300, 300), coef="temp_feel") +
    prior(exponential(.004), class="sigma"),
  family = gaussian(),
  data = bikes
)
```

Let's translate our model into brms syntax:

Model	brms syntax
$y_i \stackrel{iid}{\sim} \text{Normal}(\mu_i, \sigma)$ $\mu_i = \alpha + \beta x_i$ $\alpha \sim \text{Normal}(5000, 2000)$ $\beta \sim \text{Normal}(150, 300)$ $\sigma \sim \text{Exponential}(1/250)$	<pre>fit <- brm(rides ~ temp_feel, prior = prior(normal(5000, 2000), class="Intercept") + prior(normal(300, 300), coef="temp_feel") + prior(exponential(.004), class="sigma"), family = gaussian(), data = bikes)</pre>

That may look like a lot of code, but **brms** models can be specified more simply with the built-in defaults.

```
fit_default <- brm(rides ~ temp_feel, data = bikes)
```

- By default, **brms** assumes that **rides** is normally distributed and will use uninformative default priors that do not incorporate any domain knowledge.
- Generally in this course we will specify meaningful priors, but this can be used to approximate a default “frequentist” or maximum-likelihood estimate.
- We’ll talk more about that in a few lectures.

That may look like a lot of code, but **brms** models can be specified more simply with the built-in defaults.

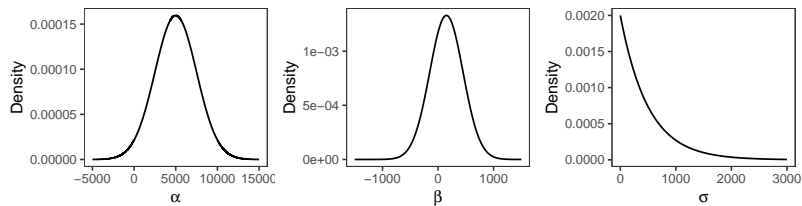
```
fit_default <- brm(rides ~ temp_feel, data = bikes)
```

- By default, **brms** assumes that **rides** is normally distributed and will use uninformative default priors that do not incorporate any domain knowledge.
- Generally in this course we will specify meaningful priors, but this can be used to approximate a default “frequentist” or maximum-likelihood estimate.
- We’ll talk more about that in a few lectures.

Plotting priors

It can be tough to come up with numbers to represent our beliefs, especially as you are getting familiarized with different probability distributions.

But it's easier if you plot things:



What do you see?

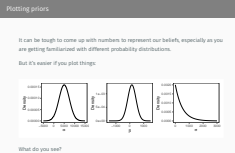
Bayesian Linear Regression

Bayesian regression in brms

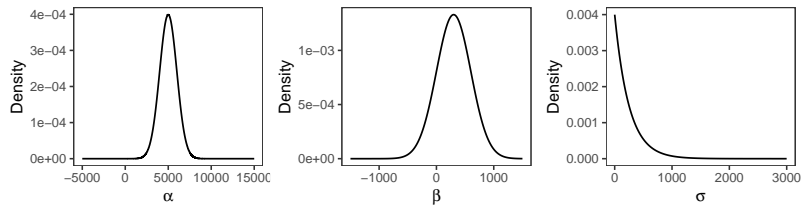
Plotting priors

2021-09-07

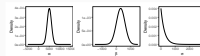
- α is allowing intercept values below zero, but we know that can't be right



- Maybe we should tighten up our prior on α
- Let's change it to $Normal(5000, 1000)$



- Maybe we should tighten up our prior on α
- Let's change it to $Normal(5000, 1000)$



- We are trying to encode our domain knowledge into priors for the parameters of our model
- Usually it is easier if we think in terms of the data we'd expect to observe, rather than parameters
- To do this we can create and visualize the prior predictive

Setting the option `sample_prior = "only"` will sample from the prior specified for the model without conditioning on the data.

```
fit_prior1 <- brm(
  rides ~ temp_feel,
  prior =
    prior(normal(5000, 1000), class="Intercept") +
    prior(normal(300, 300), coef="temp_feel") +
    prior(exponential(.004), class="sigma"),
  family = gaussian(),
  data = bikes,
  sample_prior = "only"
)
```

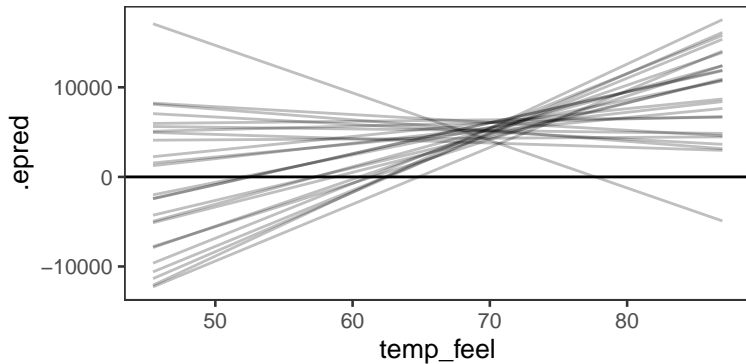
```
Setting the option sample_prior = "only" will sample from the prior specified
for the model without conditioning on the data.

fit_prior1 <- brm(
  rides ~ temp_feel,
  prior =
    prior(normal(5000, 1000), class="Intercept") +
    prior(normal(300, 300), coef="temp_feel") +
    prior(exponential(.004), class="sigma"),
  family = gaussian(),
  data = bikes,
  sample_prior = "only"
)
```

Plotting prior predictive

```
set.seed(234232)
```

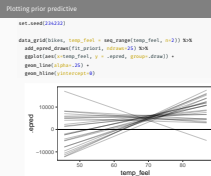
```
data_grid(bikes, temp_feel = seq_range(temp_feel, n=2)) %>%  
  add_epred_draws(fit_prior1, ndraws=25) %>%  
  ggplot(aes(x=temp_feel, y = .epred, group=.draw)) +  
  geom_line(alpha=.25) +  
  geom_hline(yintercept=0)
```



Bayesian Linear Regression

Bayesian regression in brms

Plotting prior predictive



Breaking down the code

The first line calls the `data_grid()` and `seq_along()` functions from the `modelr` package

```
data_grid(bikes, temp_feel = seq_range(temp_feel, n=2))
```

```
## # A tibble: 2 x 1
##   temp_feel
##   <dbl>
## 1      45.5
## 2      86.9
```

Bayesian Linear Regression

Bayesian regression in brms

Breaking down the code

2021-09-07

Breaking down the code

```
The first line calls the data_grid() and seq_along() functions from the modelr
package
data_grid(bikes, temp_feel = seq_range(temp_feel, n=2))

## # A tibble: 2 x 1
##   temp_feel
##   <dbl>
## 1      45.5
## 2      86.9
```

Breaking down the code (2)

Then we take that output and pipe it into the `add_epred_draws()` function from the `tidybayes` package.

```
... %>%  
  add_epred_draws(fit_prior1, ndraws=25)
```

```
## # A tibble: 6 x 6  
## # Groups:   temp_feel, .row [1]  
##   temp_feel .row .chain .iteration .draw .epred  
##       <dbl> <int>  <int>      <int> <int>  <dbl>  
## 1      45.5     1    NA         NA     1  2849.  
## 2      45.5     1    NA         NA     2  1979.  
## 3      45.5     1    NA         NA     3 -2303.  
## 4      45.5     1    NA         NA     4  9615.  
## 5      45.5     1    NA         NA     5  4598.  
## 6      45.5     1    NA         NA     6 -8097.
```

Bayesian Linear Regression

Bayesian regression in brms

Breaking down the code (2)

2021-09-07

Breaking down the code (2)

```
Then we take that output and pipe it into the add_epred_draws() function from  
the tidybayes package.  
... %>%  
  add_epred_draws(fit_prior1, ndraws=25)  
  
## # A tibble: 6 x 6  
## # Groups:   temp_feel, .row [1]  
##   temp_feel .row .chain .iteration .draw .epred  
##       <dbl> <int>  <int>      <int> <int>  <dbl>  
## 1      45.5     1    NA         NA     1  2849.  
## 2      45.5     1    NA         NA     2  1979.  
## 3      45.5     1    NA         NA     3 -2303.  
## 4      45.5     1    NA         NA     4  9615.  
## 5      45.5     1    NA         NA     5  4598.  
## 6      45.5     1    NA         NA     6 -8097.
```

And finally we plot

```
... %>%  
  ggplot(aes(x=temp_feel, y = .epred, group=.draw)) +  
  geom_line(alpha=.25) +  
  geom_hline(yintercept=0)
```

Breaking down the code (3)

- explain data_grid
- explain adding draws

And finally we plot

```
... %>%  
  ggplot(aes(x=temp_feel, y = .epred, group=.draw)) +  
  geom_line(alpha=.25) +  
  geom_hline(yintercept=0)
```

We will use several functions from the **tidybayes** package to access different types of samples from our models.

Right now we have two sources of uncertainty in our model:

1. Uncertainty in the parameters
 2. Uncertainty in the observed values
- **add_predicted_draws()**: Add draws from the prior or posterior predictive.
 - These are predicted *observations* containing both sources of uncertainty
 - **add_epred_draws()**: Add draws from the expectation of the prior/posterior predictive
 - These are the predicted values of our regression line, with only the uncertainty in the model parameters

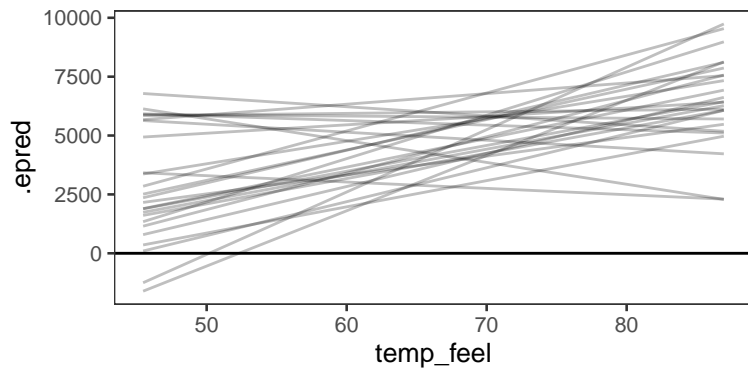
Revising the prior

This prior on β was too wide, so let's reign it in a bit. Now the prior is:

$$\alpha \sim \text{Normal}(5000, 1000)$$

$$\beta \sim \text{Normal}(100, 100)$$

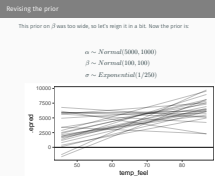
$$\sigma \sim \text{Exponential}(1/250)$$



Bayesian Linear Regression

Bayesian regression in brms

Revising the prior

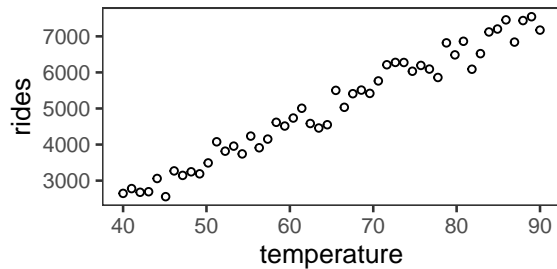


2021-09-07

Simulating the prior predictive values

The plot below shows $n = 50$ simulated observations at the expected values of our prior.

```
N <- 50; set.seed(12345)
tibble(
  temperature = seq(40, 90, length.out=N),
  rides = rnorm(N, mean = 5000 + center(temperature) * 100, sd = 250)
) %>%
  ggplot(aes(x=temperature, y = rides)) + geom_point()
```



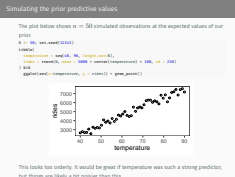
This looks too orderly. It would be great if temperature was such a strong predictor, but things are likely a bit noisier than this.

Bayesian Linear Regression

Bayesian regression in brms

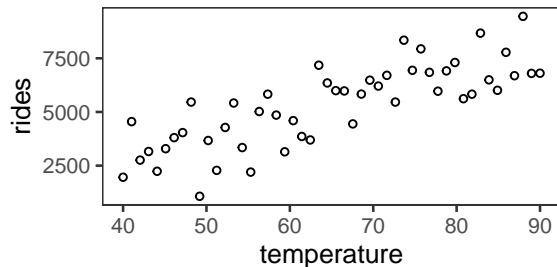
Simulating the prior predictive values

2021-09-07

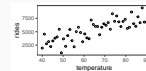


- We can also get a sense of things by simulating observations.
- It can often be a good idea to propose a prior and then spread it out a bit

Let's update the prior on σ to $Exponential(1/1000)$ and simulate some new data using our new expected value of σ .



That looks more plausible.



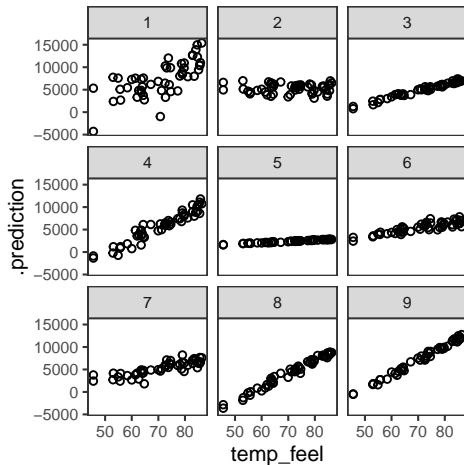
Simulating data like this can be a good way to get a quick sense of how a model functions and what kinds of data it corresponds to.

Simulating data can also be useful when we want to know whether we will be able to meaningfully estimate parameters in our statistical models, especially when models become very complex or novel

Simulated observations from the prior predictive

We can check things more formally too using the full prior predictive generated by **brms**. Here are 9 simulated datasets.

```
set.seed(2352353)
data_grid(
  bikes,
  temp_feel = sample(temp_feel, 50)
) %>%
add_predicted_draws(
  fit_prior3,
  ndraws=16
) %>%
ggplot(
  aes(
    x=temp_feel,
    y = .prediction
  )
) +
geom_point() +
facet_wrap(~.draw)
```

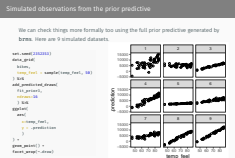


Bayesian Linear Regression

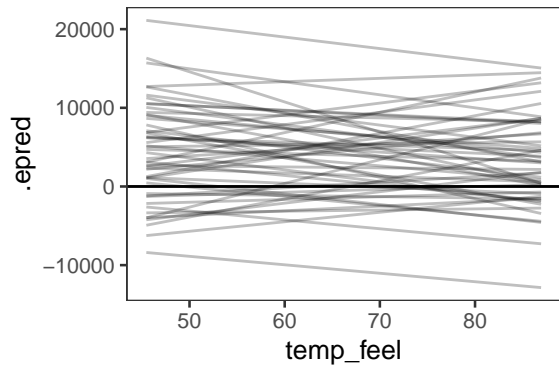
Bayesian regression in brms

Simulated observations from the prior predictive

- You can see a lot of simulated datasets have very low variance. That makes sense if you think about the shape of the prior we put on σ using an exponential distribution, the most probable value is zero.
- Here it can be ok to let one fact about the data inform us: we have a good amount of data (500 obs), and every single datapoint informs about sigma. So even if our prior is biased toward values we think are implausibly small, as long as we allow for large values, we can be pretty confident our posterior will be accurate.



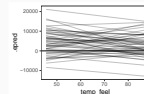
Comparison with weakly informative priors



Bayesian Linear Regression

Bayesian regression in brms

Comparison with weakly informative priors



- equally positive and negative slopes
- substantial number nearly always below zero (didn't use any prior info on intercept)

Fitting the model

2021-09-07

Bayesian Linear Regression
└ Fitting the model

Fitting the model

Now we are satisfied with our prior, we are ready to estimate the posterior of the model.

Model

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

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

$$\alpha \sim \text{Normal}(5000, 1000)$$

$$\beta \sim \text{Normal}(100, 100)$$

$$\sigma \sim \text{Exponential}(1/250)$$

brms syntax

```
fit <- brm(
  rides ~ temp_feel,
  prior =
    prior(normal(5000, 1000), class="Intercept") +
    prior(normal(100, 100), coef="temp_feel") +
    prior(exponential(.001), class="sigma"),
  family = gaussian(),
  data = bikes
)
```

Fitting the model

Fitting the model

2021-09-07

Now we are satisfied with our prior, we are ready to estimate the posterior of the model.

```
Model
fit <- brm(
  rides ~ temp_feel,
  prior =
    prior(normal(5000, 1000), class="Intercept") +
    prior(normal(100, 100), coef="temp_feel") +
    prior(exponential(.001), class="sigma"),
  family = gaussian(),
  data = bikes
)
```

- We use **brms** to ask our computer to compute a posterior probability distribution
- The estimate is made with a method called Markov Chain Monte Carlo (MCMC)
- We will leave MCMC behind an **abstraction layer**
 - **brms** does a very good job setting up MCMC, so it usually works fine
 - But there are a few diagnostics we will check to make sure things have gone smoothly

- No one can know everything.
- We will make use of an “abstraction layer”

- We use **brms** to ask our computer to compute a posterior probability distribution
- The estimate is made with a method called Markov Chain Monte Carlo (MCMC)
- We will leave MCMC behind an **abstraction layer**
 - **brms** does a very good job setting up MCMC, so it usually works fine
 - But there are a few diagnostics we will check to make sure things have gone smoothly

Before we look at our results we need to check whether the model fit properly

Three things to look out for

1. Divergent transitions: should be zero
2. \hat{R} statistic: should be 1.0 (up to 1.01 can be ok)
3. Effective sample size (ESS): should be at least about 500 for all parameters

Failing these diagnostics signals a PROBLEM. We will worry about how to fix those problems later, once we run into some.

Quick diagnostics with summary()

```
summary(fit)
```

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: rides ~ temp_feel
## Data: bikes (Number of observations: 500)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept -2176.42    355.04 -2885.35 -1490.26 1.00     3624     2783
## temp_feel   81.91      5.06   71.97   91.88 1.00     3642     2892
##
## Family Specific Parameters:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma  1283.58      39.75  1208.86  1362.75 1.00     3601     3037
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Bayesian Linear Regression

Fitting the model

Quick diagnostics with summary()

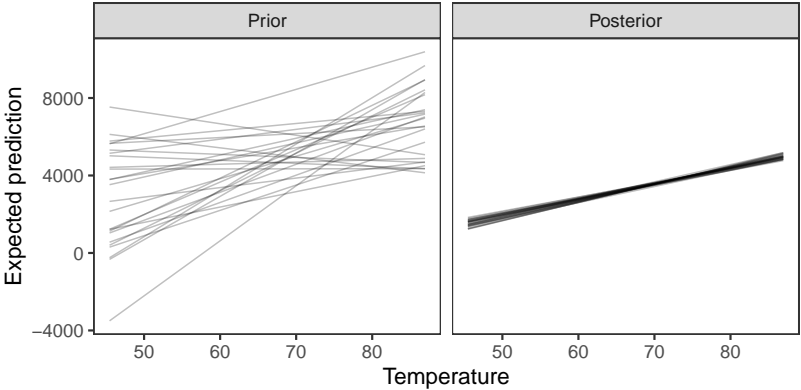
```
Quick diagnostics with summary()

summary(fit)

## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: rides ~ temp_feel
## Data: bikes (Number of observations: 500)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept -2176.42    355.04 -2885.35 -1490.26 1.00     3624     2783
## temp_feel   81.91      5.06   71.97   91.88 1.00     3642     2892
##
## Family Specific Parameters:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma  1283.58      39.75  1208.86  1362.75 1.00     3601     3037
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

2021-09-07

Posterior predictive checks

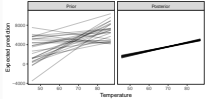


2021-09-07

Bayesian Linear Regression

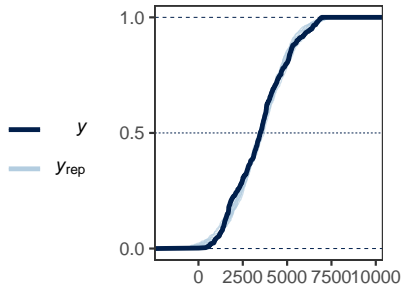
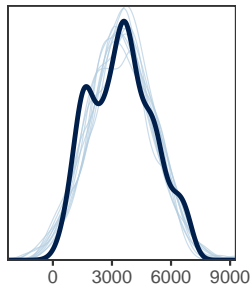
- Posterior predictive checks

- Inspecting our posterior fit



Checking the distribution

```
pp_check(fit, type="dens_overlay") +  
  pp_check(fit, type="ecdf_overlay")
```

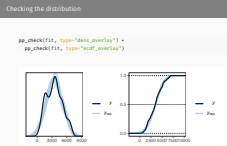


Bayesian Linear Regression

Posterior predictive checks

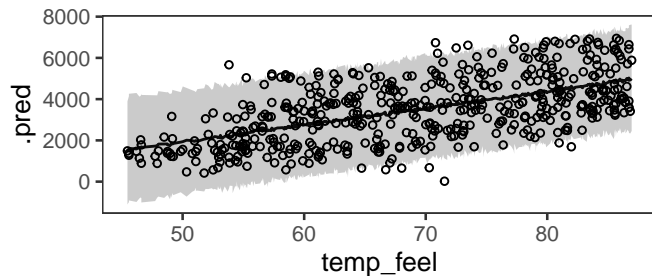
Checking the distribution

2021-09-07



Checking predictions

```
data_grid(bikes, temp_feel) %>%  
  add_predicted_draws(fit, ndraws=1000) %>% # comes out grouped by .row  
  summarize(  
    .pred = mean(.prediction),  
    .lower = quantile(.prediction, .025),  
    .upper = quantile(.prediction, .975)  
  ) %>%  
  ggplot(aes(x=temp_feel, y = .pred)) +  
  geom_line() +  
  geom_ribbon(aes(ymin=.lower, ymax=.upper), alpha=1/4) +  
  geom_point(data=bikes, aes(y=rides))
```



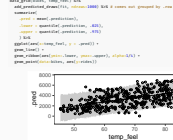
Bayesian Linear Regression

Posterior predictive checks

Checking predictions

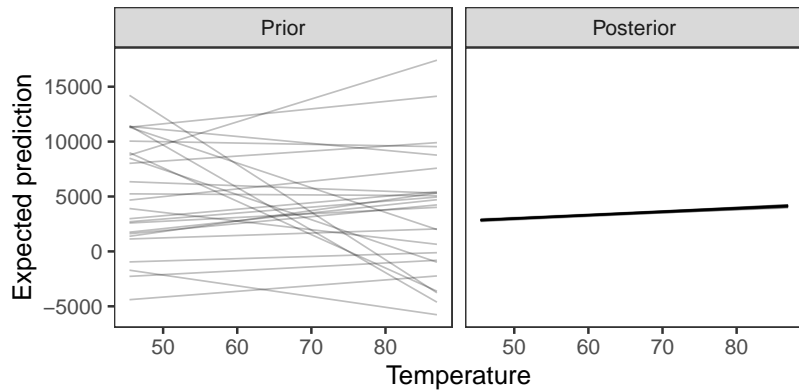
2021-09-07

Checking predictions



Weakly informative priors to posterior

- Our posterior looks the same with the weakly informative priors, so our informative priors aren't doing all that much here.
- But that's good! It means we have enough data.



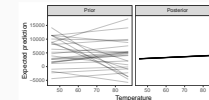
Bayesian Linear Regression

Posterior predictive checks

Weakly informative priors to posterior

Weakly informative priors to posterior

- Our posterior looks the same with the weakly informative priors, so our informative priors aren't doing all that much here.
- But that's good! It means we have enough data.



2021-09-07

Inference and prediction

2021-09-07

Bayesian Linear Regression
└ Inference and prediction

Inference and prediction

Now we are satisfied with our model, we can inspect its parameters.

```
summary(fit)
```

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: rides ~ temp_feel
## Data: bikes (Number of observations: 500)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept -2176.42    355.04 -2885.35 -1490.26 1.00     3624     2783
## temp_feel   81.91      5.06   71.97   91.88 1.00     3642     2892
##
## Family Specific Parameters:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma  1283.58      39.75  1208.86  1362.75 1.00     3601     3037
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Bayesian Linear Regression

Inference and prediction

Inference

- note that intercept values being reported to us are the uncentered values—for when the temp is 0F
- by default have 95% CI of parameters

Inference

```
Now we are satisfied with our model, we can inspect its parameters.
summary(fit)

## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: rides ~ temp_feel
## Data: bikes (Number of observations: 500)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept -2176.42    355.04 -2885.35 -1490.26 1.00     3624     2783
## temp_feel   81.91      5.06   71.97   91.88 1.00     3642     2892
##
## Family Specific Parameters:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma  1283.58      39.75  1208.86  1362.75 1.00     3601     3037
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

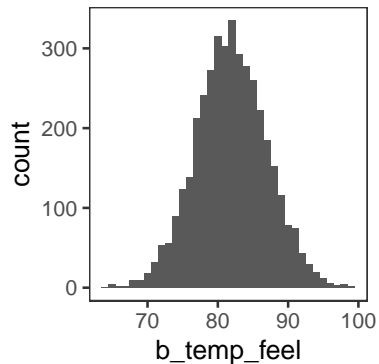
2021-09-07

Posterior parameter samples

```
posterior_samps <-  
  posterior_samples(fit) %>%  
  select(-lp__)  
head(posterior_samps)
```

##	b_Intercept	b_temp_feel	sigma
## 1	-1955.856	80.15220	1356.521
## 2	-2543.129	86.66857	1318.180
## 3	-2446.732	85.91869	1216.194
## 4	-2075.768	80.69149	1294.917
## 5	-1932.400	78.17432	1241.055
## 6	-2509.244	87.16726	1318.174

```
posterior_samps %>%  
  ggplot(aes(x=b_temp_feel)) +  
  geom_histogram(binwidth=1)
```



Bayesian Linear Regression

- Inference and prediction

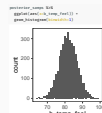
2021-09-07

- Posterior parameter samples

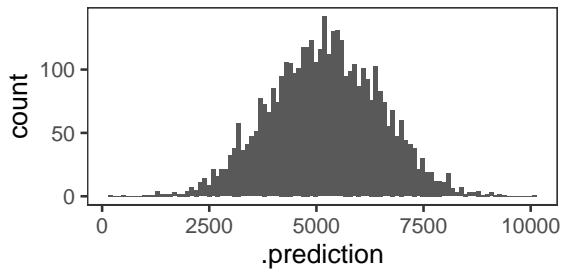
Posterior parameter samples

```
posterior_samps <-  
  posterior_samples(fit) %>%  
  select(-lp__)  
head(posterior_samps)
```

##	b_Intercept	b_temp_feel	sigma
## 1	-1955.856	80.15220	1356.521
## 2	-2543.129	86.66857	1318.180
## 3	-2446.732	85.91869	1216.194
## 4	-2075.768	80.69149	1294.917
## 5	-1932.400	78.17432	1241.055
## 6	-2509.244	87.16726	1318.174



- How many rides do we predict we'll observe on a 90° F day?



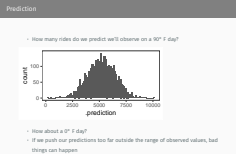
- How about a 0° F day?
- If we push our predictions too far outside the range of observed values, bad things can happen

2021-09-07

Bayesian Linear Regression

Inference and prediction

Prediction



- We need to choose a day to pull out 20% of our bikes to do maintenance
- On that day, we will not be able to support more than 5500 rides, so any rides we might have had beyond 5500 represent lost revenue, which we want to minimize
- Assume riders are charged a flat fee of \$5.00 per ride

└ Using predictions to make decisions

- this is not 100% realistic of course
- also because very few businesses are actually this data-driven

- We need to choose a day to pull out 20% of our bikes to do maintenance
- On that day, we will not be able to support more than 5500 rides, so any rides we might have had beyond 5500 represent lost revenue, which we want to minimize
- Assume riders are charged a flat fee of \$5.00 per ride

Using predictions to make decisions

- Suppose today we know that the temperature will feel like 81° F.
- What is the probability that we do not lose any revenue if we do the maintenance today?

```
post_pred <- tibble(temp_feel=81) %>%  
  add_predicted_draws(fit)
```

```
mean(post_pred$.prediction < 5500)
```

```
## [1] 0.80425
```

Bayesian Linear Regression

Inference and prediction

Using predictions to make decisions

2021-09-07

Using predictions to make decisions

```
• Suppose today we know that the temperature will feel like 81° F.  
• What is the probability that we do not lose any revenue if we do the  
  maintenance today?  
post_pred <- tibble(temp_feel=81) %>%  
  add_predicted_draws(fit)  
mean(post_pred$.prediction < 5500)  
## [1] 0.80425
```

Using predictions to make decisions

How much revenue should we expect to lose if we do the maintenance today?

```
post_pred %>%
  ungroup() %>%
  mutate(
    lost_trips = if_else(.prediction < 5500, 0, .prediction - 5500),
    lost_revenue = lost_trips*5.00
  ) %>%
  summarize(
    expected_trip_loss = mean(lost_trips),
    expected_rev_loss = mean(lost_revenue)
  )
```

```
## # A tibble: 1 x 2
##   expected_trip_loss expected_rev_loss
##           <dbl>           <dbl>
## 1             149.             744.
```

Bayesian Linear Regression

Inference and prediction

Using predictions to make decisions

2021-09-07

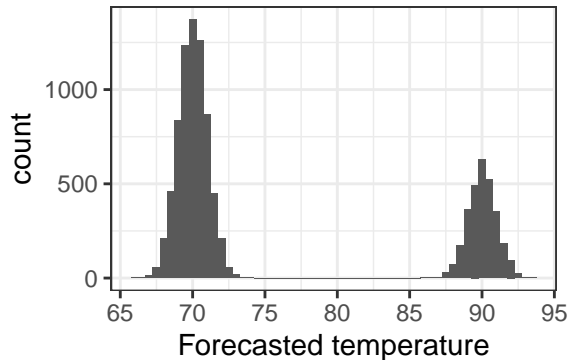
Using predictions to make decisions

```
How much revenue should we expect to lose if we do the maintenance today?
post_pred %>%
  ungroup() %>%
  mutate(
    lost_trips = if_else(.prediction < 5500, 0, .prediction - 5500),
    lost_revenue = lost_trips*5.00
  ) %>%
  summarize(
    expected_trip_loss = mean(lost_trips),
    expected_rev_loss = mean(lost_revenue)
  )

## # A tibble: 1 x 2
##   expected_trip_loss expected_rev_loss
##           <dbl>           <dbl>
## 1             149.             744.
```

Using predictions to make decisions with uncertainty

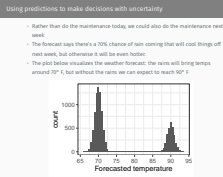
- Rather than do the maintenance today, we could also do the maintenance next week
- The forecast says there's a 70% chance of rain coming that will cool things off next week, but otherwise it will be even hotter.
- The plot below visualizes the weather forecast: the rains will bring temps around 70° F, but without the rains we can expect to reach 90° F



Bayesian Linear Regression

Inference and prediction

Using predictions to make decisions with uncertainty



2021-09-07

Using predictions to make decisions with uncertainty (2)

We compute the posterior predictive just as we did before.

```
post_pred_delay <- tibble(temp_feel = weather_samples) %>%  
  sample_n(500) %>%  
  add_predicted_draws(fit)
```

And we can compare the predictions for waiting to acting today:

```
## # A tibble: 1 x 3  
##   expected_trip_loss expected_rev_loss losses_upper_99  
##           <dbl>           <dbl>           <dbl>  
## 1           149.           744.           9829.  
  
## # A tibble: 1 x 3  
##   expected_trip_loss expected_rev_loss losses_upper_99  
##           <dbl>           <dbl>           <dbl>  
## 1           144.           721.          10612.
```

Bayesian Linear Regression

Inference and prediction

Using predictions to make decisions with uncertainty (2)

```
We compute the posterior predictive just as we did before.  
post_pred_delay <- tibble(temp_feel = weather_samples) %>%  
  sample_n(500) %>%  
  add_predicted_draws(fit)  
  
And we can compare the predictions for waiting to acting today.  
  
## # A tibble: 1 x 3  
##   expected_trip_loss expected_rev_loss losses_upper_99  
##           <dbl>           <dbl>           <dbl>  
## 1           149.           744.           9829.  
  
## # A tibble: 1 x 3  
##   expected_trip_loss expected_rev_loss losses_upper_99  
##           <dbl>           <dbl>           <dbl>  
## 1           144.           721.          10612.
```

The shape of the posterior looks normal, but that is just because of the uncertainty in the model and the forecasted weather distributions overlap sufficiently.

- We expect to lose less revenue by waiting a week
- But note that our worst-case scenario revenue losses look worse if we wait
- If these were actual costs rather than revenue, we might care more about mitigating our worst-case-scenario risk
- Working with posterior samples let's us have all this information

- We expect to lose less revenue by waiting a week
- But note that our worst-case scenario revenue losses look worse if we wait
- If these were actual costs rather than revenue, we might care more about mitigating our worst-case-scenario risk
- Working with posterior samples let's us have all this information

Exercises

2021-09-07

Bayesian Linear Regression
└ Exercises

Exercises

Explore further to see whether these priors seem reasonable by simulating some more data.

- How do things look if β is two SD above or below the mean?
- What if α is one SD lower than expected, β is two SD greater than expected, and σ is 2000?
 - What is the probability that these or more extreme conditions obtain?
 - Recall, our prior parameters are independent
 - Hint: use `rnorm()` and `rexp()` to estimate with samples, or `pnorm()` and `pexp()` to estimate directly

Explore further to see whether these priors seem reasonable by simulating some more data.

- How do things look if β is two SD above or below the mean?
- What if α is one SD lower than expected, β is two SD greater than expected, and σ is 2000?
 - What is the probability that these or more extreme conditions obtain?
 - Recall, our prior parameters are independent
 - Hint: use `rnorm()` and `rexp()` to estimate with samples, or `pnorm()` and `pexp()` to estimate directly

Extra slides

- $y_i \stackrel{iid}{\sim} \text{Normal}(\mu_i, \sigma)$
 - family = gaussian()
- $\mu_i = \alpha + \beta x_i$
 - rides ~ temp_feel
- $\alpha \sim \text{Normal}(5000, 2500)$
 - prior(normal(5000, 2000), class="Intercept")
- $\beta \sim \text{Normal}(300, 300)$
 - prior(normal(300, 300), coef="temp_feel")
- $\sigma \sim \text{Exponential}(1/250)$
 - prior(exponential(.004), class="sigma")

```

•  $y_i \stackrel{iid}{\sim} \text{Normal}(\mu_i, \sigma)$ 
• family = gaussian()
•  $\mu_i = \alpha + \beta x_i$ 
• rides ~ temp_feel
•  $\alpha \sim \text{Normal}(5000, 2500)$ 
• prior(normal(5000, 2000), class="Intercept")
•  $\beta \sim \text{Normal}(300, 300)$ 
• prior(normal(300, 300), coef="temp_feel")
•  $\sigma \sim \text{Exponential}(1/250)$ 
• prior(exponential(.004), class="sigma")
    
```