Data 582 - Bayesian Inference

Lab 4: Bayesian Linear Regression

Contents

1 Introduction

In this lab we'll be doing Bayesian Linear Regression using **rstanarm**. This package calls to the **rstan** package which takes a bit of set-up. As I mentioned at the end of last lab, I had trouble downloading the R compiler tool for MacOS and had to follow the work-around here.

1.1 Data set and packages

We start by loading in some data and neccessary pacakges.

```
# Load some packages
library(bayesrules)
library(rstanarm)
library(bayesplot) # for MCMC diagnostics plots
```

We'll be looking at the `weather_WU` dataset from the **bayesrules** package. This data comprise a sub-sample of daily weather information from the `weatherAUS` data in the **rattle** package for two Australian cities, Wollongong and Uluru.

```
data("weather_WU")
head(weather_WU)

## # A tibble: 6 x 22
##   location mintemp maxtemp rainfall windgustdir windgustspeed winddir9am winddir3pm
##   <fct>      <dbl>   <dbl>    <dbl> <ord>               <dbl> <ord>      <ord>
## 1 Uluru       12.3    30.1        0 ENE                    39 E          ENE
## 2 Uluru       20.5    35.9        5 SSE                    52 SE         SE
```

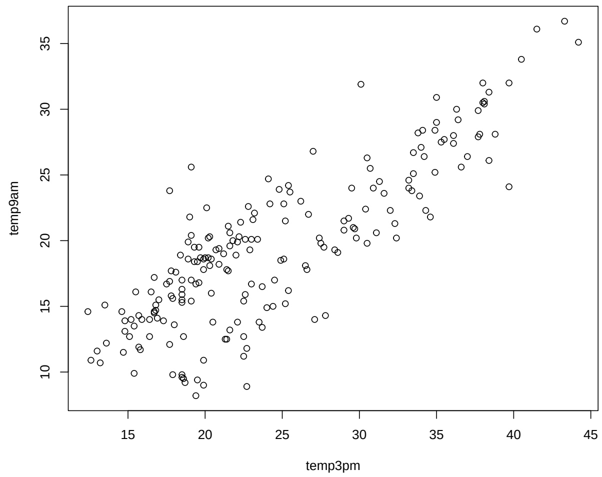
```
## 3 Uluru        15.8    41.4        0 NNW                     50 SSE        NW
## 4 Uluru        18.3    36          0 SE                      57 E          ESE
## 5 Uluru        28.9    44.8        0 ENE                     44 ESE        E
## 6 Uluru        18.1    35.9        0 ESE                     35 ESE        SE
## # ... with 14 more variables: windspeed9am <dbl>, windspeed3pm <dbl>, humidity9am <in
## #   humidity3pm <int>, pressure9am <dbl>, pressure3pm <dbl>, temp9am <dbl>,
## #   temp3pm <dbl>, raintoday <fct>, risk_mm <dbl>, raintomorrow <fct>, year <dbl>,
## #   month <dbl>, day_of_year <dbl>
```

To simplify things, we'll retain only the variables on afternoon temperatures (`temp3pm`) and a subset of possible predictors that we'd have access to in the morning:

```
library(tidyverse)
weather_WU <- weather_WU %>%
  select(location, windspeed9am, humidity9am, pressure9am, temp9am, temp3pm)
head(weather_WU)

## # A tibble: 6 x 6
##   location windspeed9am humidity9am pressure9am temp9am temp3pm
##   <fct>           <dbl>       <int>       <dbl>   <dbl>   <dbl>
## 1 Uluru              20          23       1023.    20.9    29.7
## 2 Uluru               9          71       1013.    23.4    33.9
## 3 Uluru               7          15       1012.    24.1    39.7
## 4 Uluru              28          29       1016     26.4    34.2
## 5 Uluru              24          10       1010.    36.7    43.3
## 6 Uluru              22          32       1012.    25.1    33.5
```

First we fit a simple Normal regression model of `temp3pm` with one quantitative predictor: the morning temperature `temp9am` (both measured in degrees Celsius). As you might expect, there's a positive association between these two variables – the warmer it is in the morning, the warmer it tends to be in the afternoon:

```
attach(weather_WU)
plot(temp3pm, temp9am)
```



1.2 The Model

To model this relationship, let Y_i denote the 3 p.m. temperature and X_{i1} denote the 9 a.m. temperature on a given day i. Notice that we're representing our predictor by X_{i1} here, instead of simply X_i, in order to distinguish it from other predictors used later. Then the Bayesian Normal regression model of Y by X_1 is represented by:

$$\begin{aligned}
Y_i|\beta_0, \beta_1, \sigma &\stackrel{ind}{\sim} N\left(\mu_i, \sigma^2\right) \quad \text{with} \quad \mu_i = \beta_0 + \beta_1 X_{i1} \\
\beta_{0c} &\sim N\left(25, 5^2\right) \\
\beta_1 &\sim N\left(0, 3.1^2\right) \\
\sigma &\sim \text{Exp}(0.13).
\end{aligned} \tag{11.1}$$

Consider the independent priors utilized by this model:

- β_{0c} denotes the *centered* intercept. For this particular example, the Normal prior model on the centered intercept β_{0c} reflects our prior understanding that the average afternoon temperature on a *typical* day is somewhere between 15 and 35 degrees (25±2*5). This prior specification is very useful since 0-degree mornings are rare in Australia, and hence difficult to state our prior understanding of (recall β_0 represents the typical afternoon temperature on such a 0-degree morning).

- The weakly informative priors for β_1 and σ are auto-scaled by `stan_glm()` to reflect our lack of prior information about Australian weather, as well as *reasonable* ranges for these parameters based on the simple scales of our temperature data. (see section 9.7 of the BayesRules! book)

- The fact that the Normal prior for β_1 is centered around 0 reflects a default, conservative prior assumption that the relationship between 3 p.m. and 9 a.m. temperatures might be positive ($\beta_1 > 0$), negative ($\beta_1 < 0$), or even non-existent ($\beta_1 = 0$)

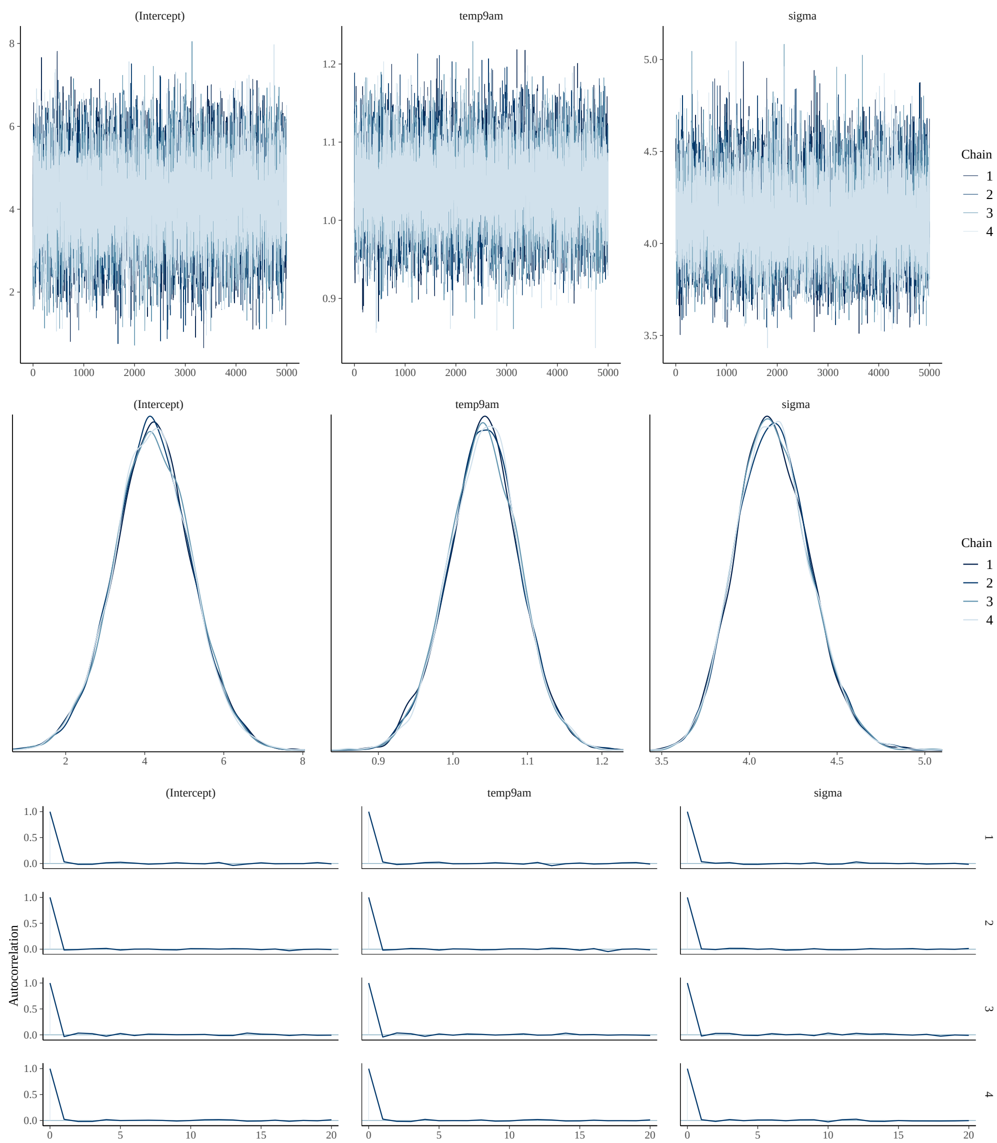
We simulate the model posterior below

```
weather_model_1 <- stan_glm(
  temp3pm ~ temp9am,
  data = weather_WU, family = gaussian,
  prior_intercept = normal(25, 5),
  prior = normal(0, 2.5, autoscale = TRUE),
  prior_aux = exponential(1, autoscale = TRUE),
  chains = 4, iter = 5000*2, seed = 84735)

##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 6.9e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.69 seco
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 1: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 1: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 1: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 1: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 1: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 1: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 1: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 1: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 1: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 1: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 1: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.126646 seconds (Warm-up)
## Chain 1:                0.192216 seconds (Sampling)
## Chain 1:                0.318862 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 1.4e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.14 seco
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 2: Iteration: 1000 / 10000 [ 10%]  (Warmup)
```

```
## Chain 2: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 2: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 2: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 2: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 2: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 2: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 2: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 2: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 2: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 2: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.124953 seconds (Warm-up)
## Chain 2:                0.208492 seconds (Sampling)
## Chain 2:                0.333445 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 1.3e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.13 seco
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 3: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 3: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 3: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 3: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 3: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 3: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 3: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 3: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 3: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 3: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 3: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.13317 seconds (Warm-up)
## Chain 3:                0.211831 seconds (Sampling)
## Chain 3:                0.345001 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 1.3e-05 seconds
```

```
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.13 seco
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 4: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 4: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 4: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 4: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 4: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 4: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 4: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 4: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 4: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 4: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 4: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.125098 seconds (Warm-up)
## Chain 4:                0.204963 seconds (Sampling)
## Chain 4:                0.330061 seconds (Total)
## Chain 4:

# Prior specification
prior_summary(weather_model_1)

## Priors for model 'weather_model_1'
## ------
## Intercept (after predictors centered)
##  ~ normal(location = 25, scale = 5)
##
## Coefficients
##   Specified prior:
##     ~ normal(location = 0, scale = 2.5)
##   Adjusted prior:
##     ~ normal(location = 0, scale = 3.1)
##
## Auxiliary (sigma)
##   Specified prior:
##     ~ exponential(rate = 1)
##   Adjusted prior:
##     ~ exponential(rate = 0.13)
## ------
## See help('prior_summary.stanreg') for more details

# MCMC diagnostics
```

```r
mcmc_trace(weather_model_1, size = 0.1)
mcmc_dens_overlay(weather_model_1)
mcmc_acf(weather_model_1)
neff_ratio(weather_model_1)

## (Intercept)     temp9am       sigma
##     0.99710     1.00255     0.94530

rhat(weather_model_1)

## (Intercept)     temp9am       sigma
##   0.9999352   0.9999668   0.9999059
```



To do prediction we could use:

```r
# Posterior credible intervals
posterior_interval(weather_model_1, prob = 0.80)

##                   10%      90%
## (Intercept) 2.9498083 5.448752
## temp9am     0.9802648 1.102423
## sigma       3.8739305 4.409474
```