

Assignment 2

Julia Haaf & Nicole Cruz

```
library(brms)

## Loading required package: Rcpp
## Loading 'brms' package (version 2.22.0). Useful instructions
## can be found by typing help('brms'). A more detailed introduction
## to the package is available through vignette('brms_overview').

##
## Attaching package: 'brms'
## The following object is masked from 'package:stats':
## 
##     ar
library(ggplot2)
```

Exercise 1: Check for parameter recovery in a linear model using simulated data.

Generate some simulated independent and identically distributed data with $n = 100$ data points as follows:

```
y <- rnorm(100, mean = 500, sd = 50)
```

Next, fit a simple linear model with a normal likelihood:

$$y_n \sim Normal(\mu, \sigma) \quad (1)$$

Specify the following priors:

$$\begin{aligned} \mu &\sim Uniform(0, 60000) \\ \sigma &\sim Uniform(0, 2000) \end{aligned} \quad (2)$$

Generate posterior predictive distributions of the parameters and check that the true values of the parameters $\mu = 500, \sigma = 50$ are recovered by the model. What this means is that you should check whether these true values lie within the range of the posterior distributions of the two parameters. This is a good sanity check for finding out whether a model can in principle recover the true parameter values correctly.

Solution

Generate some simulated data with known parameter values:

```
y<-rnorm(100,mean=500,sd=50)
sim_dat<-data.frame(y=y)
```

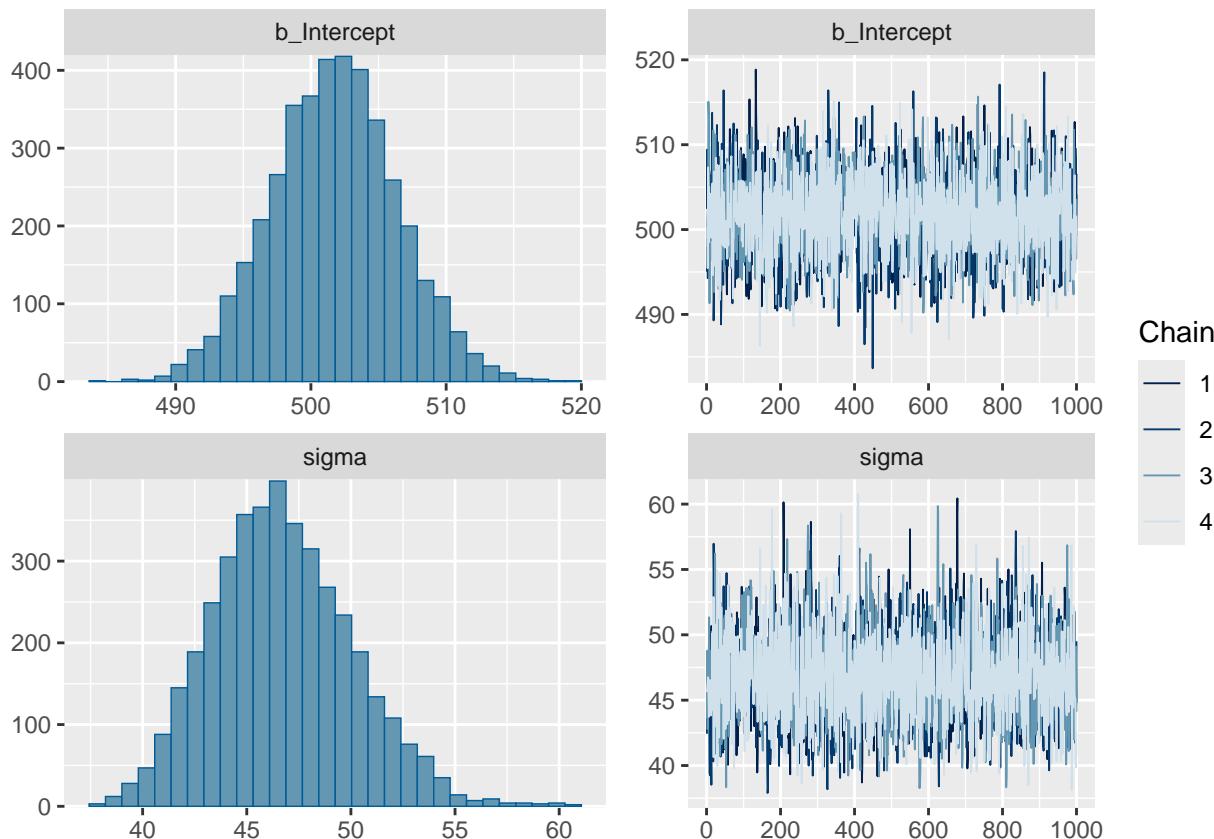
```

sim_model <- brm(y ~ 1,
  data = sim_dat,
  family = gaussian(),
  prior = c(
    prior(uniform(0, 60000), class = Intercept, lb = 0, ub = 60000),
    prior(uniform(0, 2000), class = sigma, lb = 0, ub = 2000)
  ),
  chains = 4,
  iter = 2000,
  warmup = 1000
)

```

Look at the posterior distributions:

```
plot(sim_model)
```



Confirm through visual inspect that the true values ($\mu = 500, \sigma = 50$) do fall within the respective posterior distributions shown in the plot.

Exercise 2: Chocolate Bars

Imagine your favorite chocolate bar is called *Nutty Buddy Bites*. Because you buy Nutty Buddies so often, you notice that they seem to be lighter than just a few years ago even though the weight on the package hasn't changed. As a good statistician, you decide to test whether your suspicions are warranted. For the sake of research, you buy 100 Nutty Buddies and a good scale, and you weigh each of the chocolate bars. You save the final data set as `nutties.csv`.

- The packaging says that the bars are supposed to weigh 50g. Using normal distributions, choose priors

that represent **your** assumptions/beliefs about chocolate bar weight. To think about a reasonable set of priors for μ and σ , you should come up with your own subjective assessment about what you think a reasonable range of values can be for μ and how much variability might happen. There is no correct answer here.

- b. Fit the model with just a few iterations, say 50 iterations (set warmup to the default of 25, and use four chains). Does the model converge?
- c. Now fit the model properly with sufficient iterations. Visualize the posteriors.

Solution

a.

```
nuttiesdat <- read.csv("data/nutties.csv")

nuttiesprior <- c(brms::prior(normal(50, 2)
                                , coef = Intercept)
                  , brms::prior(normal(3, 3)
                                , class = sigma))
```

b.

```
fit <- brm(data = nuttiesdat
            , weight ~ 0 + Intercept
            , prior = nuttiesprior
            , iter = 50
            , warmup = 25
            , silent = 2
            , refresh = 0)

## Trying to compile a simple C file
## Running /usr/lib/R/bin/R CMD SHLIB foo.c
## using C compiler: 'gcc (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0'
## gcc -I"/usr/share/R/include" -DNDEBUG -I"/home/juliahaaf/R/x86_64-pc-linux-gnu-library/4.4/Rcpp/include"
## In file included from /home/juliahaaf/R/x86_64-pc-linux-gnu-library/4.4/RcppEigen/include/Eigen/Core
##                 from /home/juliahaaf/R/x86_64-pc-linux-gnu-library/4.4/RcppEigen/include/Eigen/Dense
##                 from /usr/lib/R/site-library/StanHeaders/include/stan/math/prim/fun/Eigen.hpp:22,
##                     from <command-line>:
## /home/juliahaaf/R/x86_64-pc-linux-gnu-library/4.4/RcppEigen/include/Eigen/src/Core/util/Macros.h:679
##   679 | #include <cmath>
##       | ^~~~~~
## compilation terminated.
## make: *** [/usr/lib/R/etc/Makeconf:195: foo.o] Error 1

## Warning: There were 4 chains where the estimated Bayesian Fraction of Missing Information was low. See
## https://mc-stan.org/misc/warnings.html#bfmi-low

## Warning: Examine the pairs() plot to diagnose sampling problems
## Warning: The largest R-hat is 2.67, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#r-hat

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#bulk-ess
```

```

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quantiles
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#tail-ess

c.

fit <- brm(data = nuttiesdat
            , weight ~ 0 + Intercept
            , prior = nuttiesprior
            , iter = 2000
            , warmup = 500
            , silent = 2
            , refresh = 0
            , cores = 4)

## Trying to compile a simple C file

## Running /usr/lib/R/bin/R CMD SHLIB foo.c
## using C compiler: 'gcc (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0'
## gcc -I"/usr/share/R/include" -DNDEBUG -I"/home/juliahaaf/R/x86_64-pc-linux-gnu-library/4.4/Rcpp/include"
## In file included from /home/juliahaaf/R/x86_64-pc-linux-gnu-library/4.4/RcppEigen/include/Eigen/Core
##                 from /home/juliahaaf/R/x86_64-pc-linux-gnu-library/4.4/RcppEigen/include/Eigen/Dense
##                 from /usr/lib/R/site-library/StanHeaders/include/stan/math/prim/fun/Eigen.hpp:22,
##                 from <command-line>:
## /home/juliahaaf/R/x86_64-pc-linux-gnu-library/4.4/RcppEigen/include/Eigen/src/Core/util/Macros.h:679
##   679 | #include <cmath>
##       | ^~~~~~
## compilation terminated.
## make: *** [/usr/lib/R/etc/Makeconf:195: foo.o] Error 1

fit

## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: weight ~ 0 + Intercept
## Data: nuttiesdat (Number of observations: 100)
## Draws: 4 chains, each with iter = 2000; warmup = 500; thin = 1;
##        total post-warmup draws = 6000
##
## Regression Coefficients:
##             Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     48.15      0.14    47.89    48.43 1.00      5092     3478
## 
## Further Distributional Parameters:
##             Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma        1.35      0.10     1.18     1.57 1.00      4515     3200
## 
## Draws were sampled 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).

```

Exericse 3: Other priors

- Can you come up with very informative priors that influence the posterior in a noticeable way (use normal distributions for priors, not uniform priors)? Again, there are no correct answers here; you may have to try several different priors before you can noticeably influence the posterior.

- b. Generate and plot prior predictive distributions based on this prior and plot them.
- c. Generate posterior predictive distributions based on this prior and plot them.

Solution

a.

```

nuttiesprior2 <- c(brms::prior(normal(50, 0.1)
                                , coef = Intercept)
                                , brms::prior(normal(3, 3)
                                , class = sigma))

fit2 <- brm(data = nuttiesdat
             , weight ~ 0 + Intercept
             , prior = nuttiesprior2
             , iter = 2500
             , warmup = 1000
             , silent = 2
             , refresh = 0
             , cores = 4)

## Trying to compile a simple C file
## Running /usr/lib/R/bin/R CMD SHLIB foo.c
## using C compiler: 'gcc (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0'
## gcc -I"/usr/share/R/include" -DNDEBUG -I"/home/juliahaaf/R/x86_64-pc-linux-gnu-library/4.4/Rcpp/include"
## In file included from /home/juliahaaf/R/x86_64-pc-linux-gnu-library/4.4/RcppEigen/include/Eigen/Core
##                 from /home/juliahaaf/R/x86_64-pc-linux-gnu-library/4.4/RcppEigen/include/Eigen/Dense
##                 from /usr/lib/R/site-library/StanHeaders/include/stan/math/prim/fun/Eigen.hpp:22,
##                     from <command-line>
## /home/juliahaaf/R/x86_64-pc-linux-gnu-library/4.4/RcppEigen/include/Eigen/src/Core/util/Macros.h:679
##   679 | #include <cmath>
##   |           ^
## compilation terminated.
## make: *** [/usr/lib/R/etc/Makeconf:195: foo.o] Error 1

fit2

## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: weight ~ 0 + Intercept
## Data: nuttiesdat (Number of observations: 100)
## Draws: 4 chains, each with iter = 2500; warmup = 1000; thin = 1;
##        total post-warmup draws = 6000
##
## Regression Coefficients:
##             Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    49.63      0.10    49.42    49.83 1.00     3049     3441
## 
## Further Distributional Parameters:
##             Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma       2.02      0.17     1.72     2.37 1.00     2829     3403
## 
## Draws were sampled 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).

```

b.

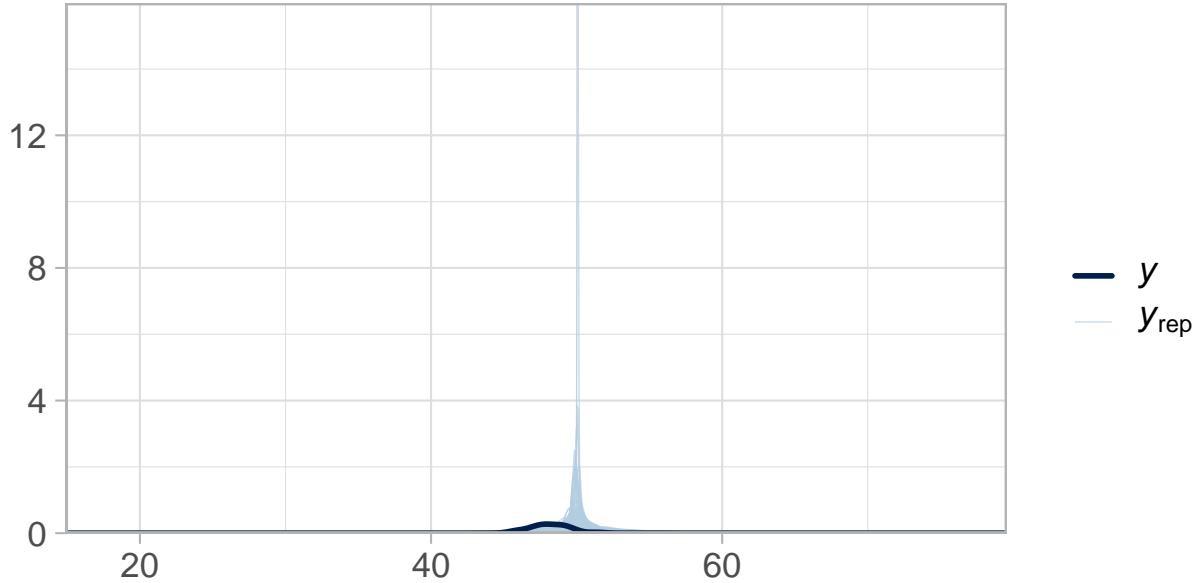
```
prior2 <- brm(data = nuttiesdat
  , weight ~ 0 + Intercept
  , prior = nuttiesprior2
  , sample_prior = "only"
  , iter = 2500
  , warmup = 1000
  , silent = 2
  , refresh = 0
  , cores = 4)

## Trying to compile a simple C file

## Running /usr/lib/R/bin/R CMD SHLIB foo.c
## using C compiler: 'gcc (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0'
## gcc -I"/usr/share/R/include" -DNDEBUG -I"/home/juliahaaf/R/x86_64-pc-linux-gnu-library/4.4/Rcpp/include"
## In file included from /home/juliahaaf/R/x86_64-pc-linux-gnu-library/4.4/RcppEigen/include/Eigen/Core
##                 from /home/juliahaaf/R/x86_64-pc-linux-gnu-library/4.4/RcppEigen/include/Eigen/Dense
##                 from /usr/lib/R/site-library/StanHeaders/include/stan/math/prim/fun/Eigen.hpp:22,
##                 from <command-line>:
## /home/juliahaaf/R/x86_64-pc-linux-gnu-library/4.4/RcppEigen/include/Eigen/src/Core/util/Macros.h:679
##   679 | #include <cmath>
##   |     ^
## compilation terminated.
## make: *** [/usr/lib/R/etc/Makeconf:195: foo.o] Error 1

prior_draws <- posterior_predict(prior2)

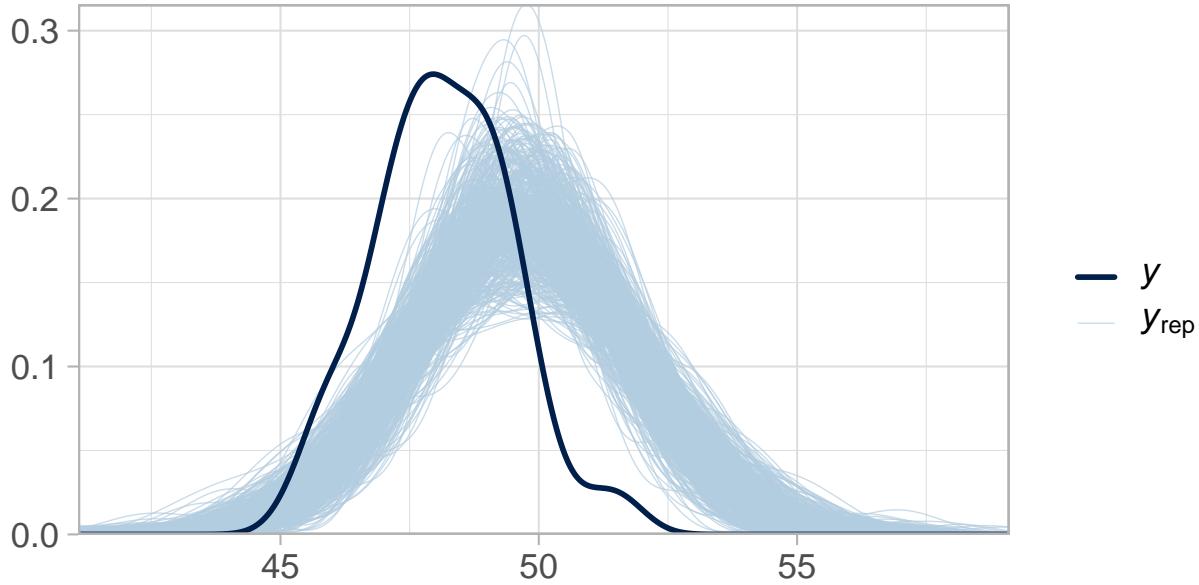
bayesplot::ppc_dens_overlay(nuttiesdat$weight, prior_draws[1:500,]) +
  theme_light(base_size = 16)
```



c.

```
post_draws <- posterior_predict(fit2)

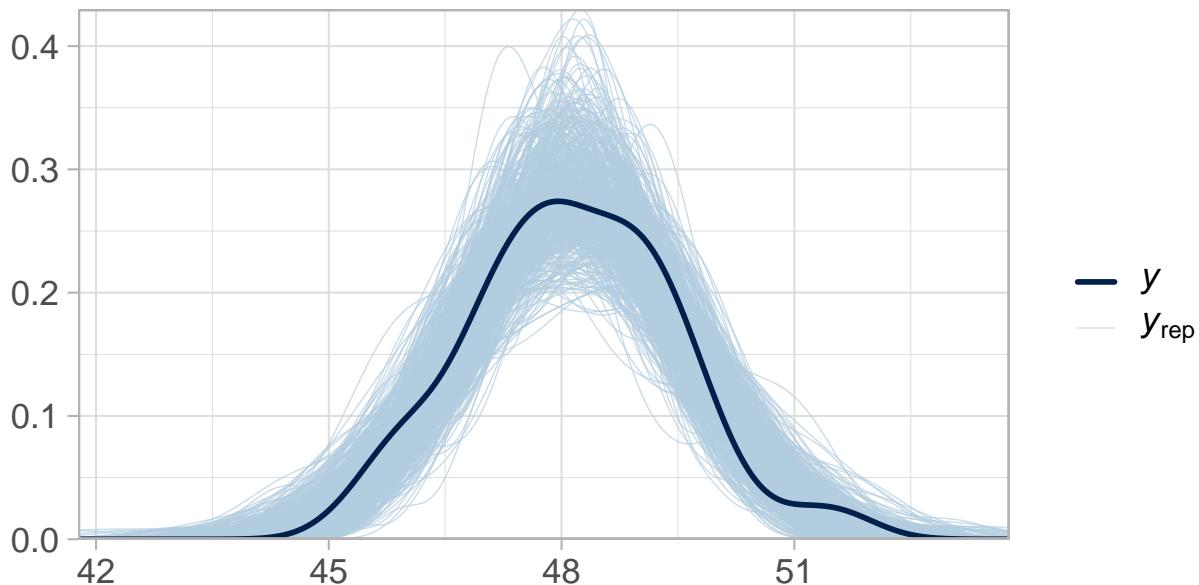
bayesplot::ppc_dens_overlay(nuttiesdat$weight, post_draws[1:500,]) +
  theme_light(base_size = 16)
```



In comparison, let's check the original priors:

```
post_draws <- posterior_predict(fit)

bayesplot::ppc_dens_overlay(nuttiesdat$weight, post_draws[1:500,]) +
  theme_light(base_size = 16)
```



Exercise 4: Your visualization set

Generate a script with some useful visualizations for your personal Bayesian workflow. Include Prior and Posterior prediction, visualization of the posterior distributions, and anything else you find useful. Also note which visualizations can be reused on a regular basis and which depend a lot on the model/data at hand.