

Assignment 7

Hierarchical model in Stan

anonymous

1 General information

This is the template for [assignment 7](#). You can download the [separate model with bad priors](#) and the [qmd-file](#) or copy the code from this rendered document after clicking on `</> Code` in the top right corner.

Please replace the instructions in this template by your own text, explaining what you are doing in each exercise.

2 Hierarchical Model: Chicken Data with Stan (6p)

2.1 Choosing a weakly informative prior by intuition

2.1.1 (a)

2.1.2 (b)

2.1.3 (c)

2.1.4 (d)

2.1.5 (e)

2.2 Choosing a weakly informative prior using external references

2.2.1 (f)

2.2.2 (g)

2.2.3 (h)

2.2.4 (i)

2.3 Non-normal priors

2.3.1 (j)

2.4 Modeling diet effects on chicken weight

! Data inside, don't peek before you have set your priors!

! Have you set your priors?

```
data("ChickWeight")

Chick12 <- ChickWeight |> filter(Time == 12)

head(Chick12)
```

Grouped Data: weight ~ Time | Chick

	weight	Time	Chick	Diet
1	106	12	1	1
2	122	12	2	1
3	115	12	3	1
4	102	12	4	1
5	141	12	5	1
6	141	12	6	1

2.4.1 (k)

2.4.2 (l)

For the figures below, we use the earlier samples for the separate model with bad priors. When you have implemented the pooled and hierarchical models, edit the code below to include samples from your model into the figures.

2.4.3 (m)

```
ggplot(posterior_mean_diet_4, aes(x = mean_diet_4, y = model_name)) +  
  stat_dotsinterval(quantiles = 100, scale = .9) +  
  vline_at(diet_means[4], size = 1, linetype = "dashed") +  
  # Annotate the vline from above.  
  annotate("text", label = "Observation mean", x = diet_means[4] - 5, y = .7,  
    hjust = "right", size = 6) +  
  # Add title and axis labels. One line to make everything so much more clear!  
  labs(  
    title = "Mean of diet 4",  
    x = "Weight (g)",  
    y = "Model"  
  )
```

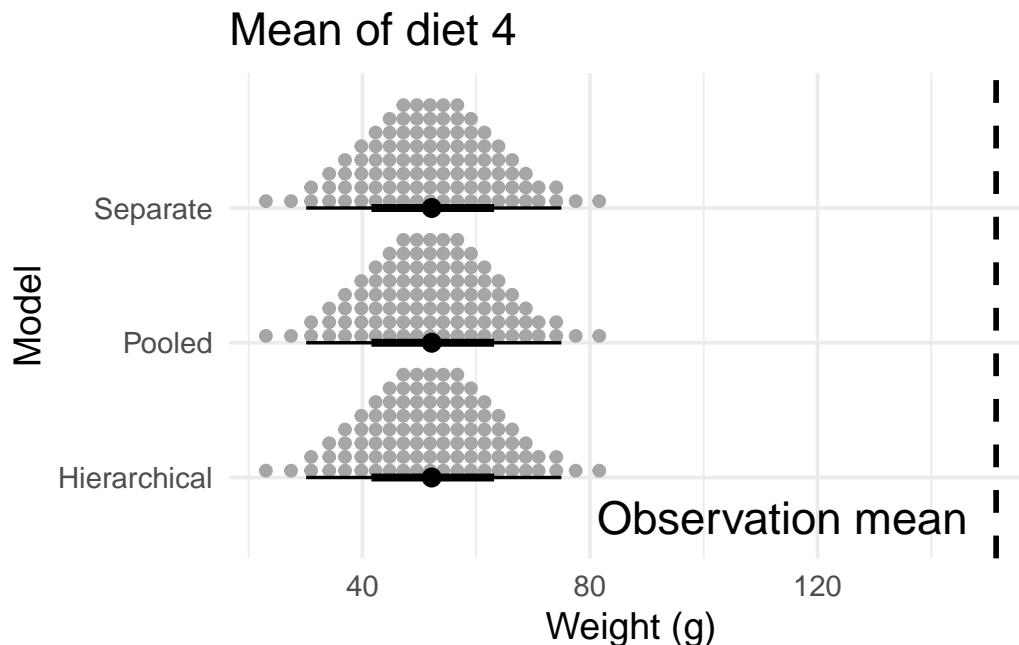


Figure 1: Posterior distribution of the mean weight of chicks consuming diet 4.

2.4.4 (n)

```
ggplot(predicted_weight_diet_4, aes(x = predicted_weight, y = model_name)) +  
  stat_dotsinterval(quantiles = 100, scale = .9) +  
  vline_at(diet_means[4], size = 1, linetype = "dashed") +  
  # Annotate the vline from above.
```

```

annotate("text", label = "Observation mean", x = diet_means[4] - 5, y = .7,
        hjust = "right", size = 6) +
# Add title and axis labels. One line to make everything so much more clear!
labs(
  title = "Weigth of a chick with diet 4",
  x = "Weight (g)",
  y = "Model"
)

```

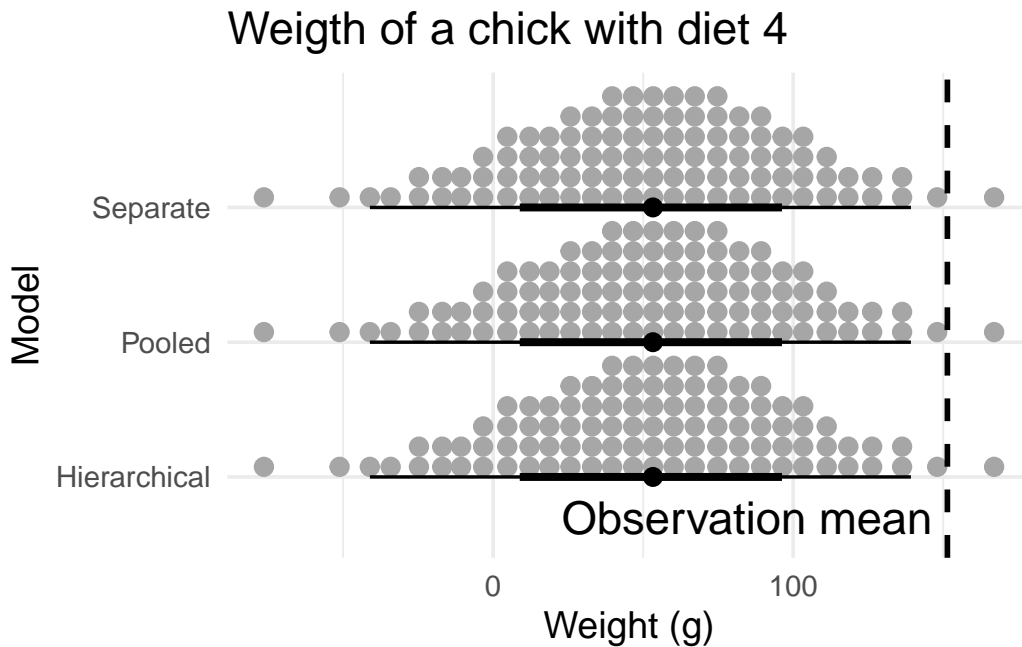


Figure 2: The (posterior) predictive distribution of the weigth of a chick consuming diet 4.

2.4.5 (o)

```

ggplot(posterior_mean_diet_5, aes(x = mean_diet_5, y = model_name)) +
# Draw the mean of each diet from the data as a dashed vertical line.
vline_at(diet_means, size = .5, linetype = "dashed") +
# dotsinterval gives mean, 50%, and 90% intervals + dotsplot with each dot
# representing 1% of data (quantiles = 100).
stat_dotsinterval(quantiles = 100, scale = .9) +
# Annotate the vline from above.
annotate(geom = "text", label = "Means of observed diets", y = .7, x = 100,
        hjust = "right", size = 5, family = "sans") +
# Add title and axis labels. One line to make everything so much more clear!
labs(title = "Mean of a new diet",
  x = "Weight (g)",
  y = "Model")

```

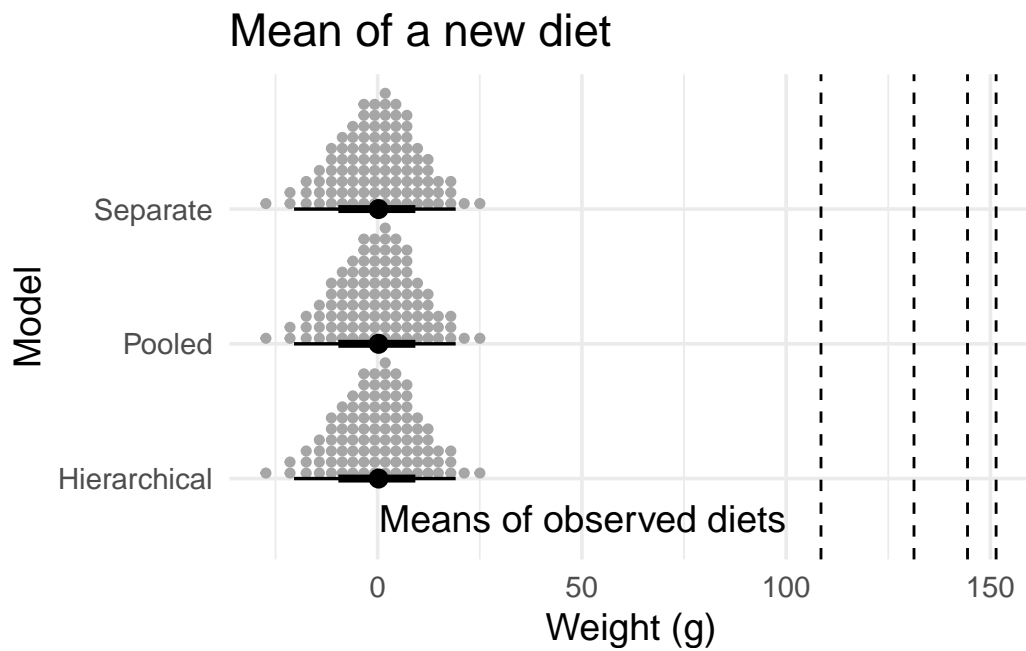


Figure 3: Posterior distribution of the mean weight of chicks consuming the new diet 5 not seen before.

2.4.6 (p)

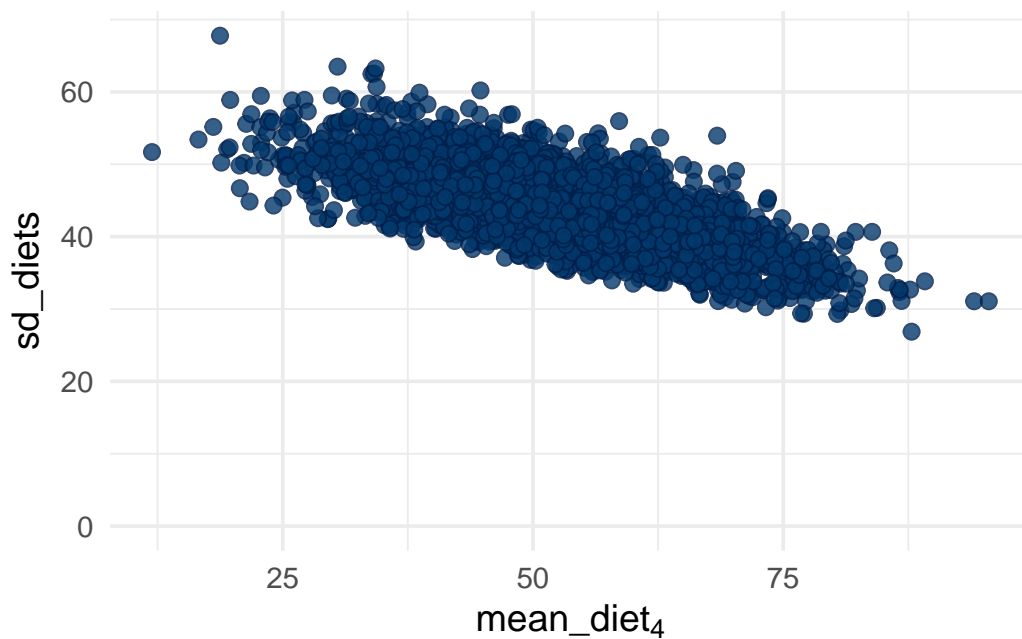
3 Hierarchical model with BRMS (3p)

3.0.1 (a)

```
bayesplot::mcmc_scatter(x = fit_hierarchical$draws(variables = c("mean_diet[4]", "sd_diets")),
                        np = nuts_params(fit_hierarchical)) +
  scale_y_log10() +
  labs(x = expression(mean_diet[4]), y = expression(sd_diets)) +
  ylim(c(0, NA))
```

Scale for y is already present.

Adding another scale for y, which will replace the existing scale.



3.0.2 (b)

Because `brms` is a bit chatty, suppress it's output in the PDF using the block above, but copy the code you executed into the code block below, which doesn't execute:

```
# Copy the you used to fit the brms model here
```

3.0.3 (c)

```
# Draws for mu_4
mu_4 = posterior_epred(brms_fit, newdata = data.frame(Diet=4))

# Compute the mean, and quantiles. Remember to round your answers accordingly.
# ...
```

3.0.4 (d)

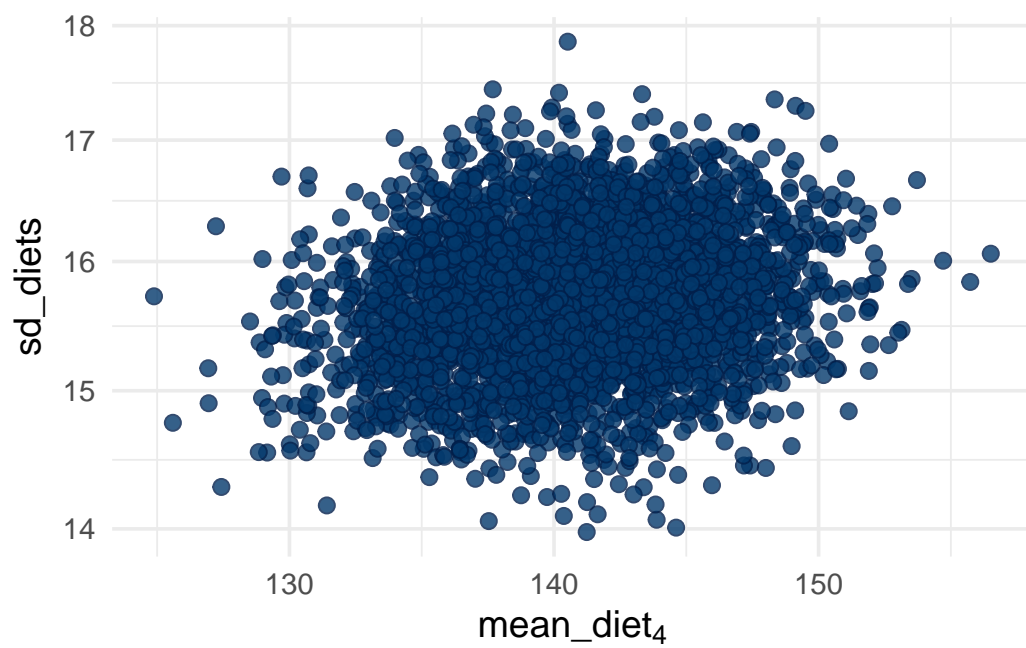
3.0.5 Scatterplot for non-centered parametrization

Due the non-centered parametrization, we need to transform compute the μ_d term as the sum of the population intercept and the group specific deviation from the intercept. You can choose which diet to plot by modifying the `d` integer in `r_Diet[d,Intercept]`.

```
draws = as_draws_df(brms_fit) |>
  posterior::mutate_variables(mean_diet_4 = `r_Diet[4,Intercept]` + b_Intercept)

bayesplot::mcmc_scatter(draws,
  pars = c("mean_diet_4", "sd_Diet__Intercept"),
  np = nuts_params(brms_fit)) +
```

```
scale_y_log10() +  
xlab(expression(mean_diet[4])) +  
ylab(expression(sd_diets))
```



3.0.6 (e)