

Assignment 7

Hierarchical model in Stan

anonymous

1 General information

I did not use AI for solving this exercise.

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

2.1 Choosing a weakly informative prior by intuition

2.2 (a)

My rationale for this estimation is as follows:

A chicken typically hatches from an egg with an initial weight of around 40 grams. The average daily weight gain is not uniform for each week and varies considerably depending on age

If we consider 21 days to become fully grown: During this growth period, it's natural for a chicken to experience variations in its weight, with an average daily weight gain of about 1.5 to 16.5 grams. This range accounts for fluctuations due to factors like diet, genetics, and environmental conditions. Considering this weight gain range over the course of 21 days, a fully grown chicken's weight is expected to fall within the broader range of 70 to 380 grams.

If we consider 18 weeks to become fully grown: During this growth period, it's natural for a chicken to experience variations in its weight, with an average daily weight gain of about 15.5 to 31.5 grams. This range accounts for fluctuations due to factors like diet, genetics, and environmental conditions. Considering this weight gain range over the course of 18×7 days, a fully grown chicken's weight is expected to fall within the broader range of 2000 to 4000 grams.

2.3 (b)

Considering the growth period of the first 12 days in a chicken's life, the rate of weight gain is typically lower than during the full growth period. On average, a 12-day old chicken experiences variations in its weight, with an average daily weight gain of approximately 1 to 14.5 grams. If we consider a newly hatched chicken with an initial weight of around 40 grams, it is reasonable to estimate that a 12-day old chicken's weight would fall within the range of approximately 52 to 220 grams. I would choose a mean μ_0 for the weakly informed prior of the parameter μ to be around 136 grams. Note that the average daily weight gain during first 12 days is less than first 21 days with a small difference and it is less than first 18 weeks with a big difference.

2.4 (c)

During the first 12 days of a chicken's life, we can define biologically grounded constraints on its weight range. These constraints are rooted in the natural growth and development of the chicken. Firstly, it's reasonable to assume that a chicken might experience a slight weight loss on some days, which could be up to 1 gram. Considering that it was born with a weight of 40 grams, it's plausible for a 12-day old chick to have a minimum weight of approximately 30 grams.

Conversely, it's essential to acknowledge that the rate of weight gain during this early stage is more moderate compared to the full growth period. To provide a conservative estimate, we should consider that the daily weight gain is unlikely to exceed 22 grams in a day. This upper limit aligns with the idea that, during a full growth period, the maximum daily gain is around 16.5 grams. A limit of 22 grams significantly surpasses this rate, allowing for reasonable fluctuations in daily weight gain.

Considering these considerations, we can define a conservative weight range for any 12-day old chick, spanning from 30 to 300 grams.

2.5 (d)

Using the formula for standard deviation:

$$\sigma_{plausible} = \frac{\text{Maximum} - \text{Minimum}}{6}$$

and given that the Maximum is 300 and the Minimum is 30, we can calculate the standard deviation as follows:

$$\sigma_{plausible} = \frac{300 - 30}{6} = \frac{270}{6} = 45 \text{ grams}$$

So, based on this formula, the plausible sd for the prior would be 45 grams. Therefore, the standard deviation is $\sigma_0 = 450$.

2.6 (e)

The prior for the mean weight μ , given the standard deviation $\sigma_0 = 450$ and mean $\mu_0 = 136$, can be expressed as a normal distribution:

$$\mu \sim \mathcal{N}(\mu_0, \sigma_0)$$

The final prior in mathematical notation would be:

$$\mu \sim \mathcal{N}(136, 450)$$

2.7 Choosing a weakly informative prior using external references

2.8 (f)

This is my references as a external references:

“[First one](#)” and “[Second one](#)”

I this sources, which states that the weight range of fully grown farm chickens is typically between 2.5 kilograms and 4 kilograms. To adjust this range for a 12-day old chick, I would adjust that a 12-day old chick gain weight 6.6 grams to 16.6 grams which is significantly less than average growth rate of a fully grown chicken. So, I would estimate the weight range for a 12-day old chick to be between 120 grams and 240 grams, which is a reasonable adjustment.

2.9 (g)

Based on this reference range, I would choose a mean of $\mu_0 = 180$ grams for our weakly informed prior.

2.10 (h)

For the upper bound: $Pr(180 + 3\sigma < 240) \approx 0.997$, solving for σ .

Using the z-score formula:

$$z = \frac{X - \mu}{\sigma}$$

where X is the value you want to find the probability for (in this case, 240), μ is the mean (average) of the distribution, and σ is the standard deviation.

First, find the z-score for the probability 0.997 using a standard normal distribution table or calculator: $z \approx 2.7$.

Now, you can set up the equation:

$$2.7 = \frac{240 - \mu}{\sigma}$$

Solve for σ :

$$\sigma_{\text{upper}} = \frac{240 - \mu}{2.7} \approx 22$$

For the lower bound we should look to negative z-score. Find the z-score for the probability $1 - 0.997$ using a standard normal distribution table or calculator: $z \approx -2.7$.

Now, set up the equation:

$$-2.7 = \frac{120 - \mu}{\sigma}$$

Solve for σ :

$$\sigma_{\text{lower}} = \frac{120 - \mu}{-2.7} \approx 22$$

2.11 (i)

The prior for the mean weight μ , given the standard deviation $\sigma_0 = 22$ and mean $\mu_0 = 180$, can be expressed as a normal distribution:

$$\mu \sim \mathcal{N}(\mu_0, \sigma_0)$$

The final prior in mathematical notation would be:

$$\mu \sim \mathcal{N}(180, 22)$$

2.12 Non-normal priors

2.13 (j)

Non-negative: When dealing with variables that cannot have negative values, like weight, height, or age, using a normal distribution as a prior is not ideal because it assigns nonzero probabilities to negative values. In such situations, it might be more suitable to opt for a distribution that exclusively encompasses non-negative values, such as a Gamma or Log-normal distribution.

Bounded Variables: When dealing with variables that have strict bounds and cannot take on values outside of those bounds, a normal distribution as a prior may not be appropriate. For example, if you are modeling a percentage or proportion that is bounded between 0 and 1, a normal distribution could assign non-zero probabilities to values outside this range, which is not meaningful.

The key is to choose a prior distribution that aligns with the nature and constraints of the variable you are modeling. Normal distributions are often used for variables that are continuous and unbounded. Moreover we can note that noncontinuous variables is not suitable for normal distribution.

2.14 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

2.15 (k)

1. Separate model:

The mathematical formulation of the model is:

$$\begin{aligned}\mu_d &\sim N(180, 22) \\ \sigma_d &\sim \text{exponential}(0.02) \\ w_{i,d} &\sim N(\mu_d, \sigma_d)\end{aligned}$$

Unlike other models, our approach assigns unique parameters, σ_j and μ_j , to each diet j . Each of these parameters is associated with its own distinct prior distribution. This modeling choice reflects our underlying assumption that the diets are independent of each other.

2. Pooled model:

The mathematical formulation of the model is:

$$\begin{aligned}\mu &\sim N(180, 22) \\ \sigma &\sim \text{exponential}(0.02) \\ w_i &\sim N(\mu, \sigma)\end{aligned}$$

Unlike other models, our approach employs a common set of parameters, σ and μ , shared by all diets. Each of these parameters is associated with its own specific prior distribution. This modeling choice signifies our underlying assumption that all measurements are pooled together, and there is no differentiation between diets.

3. Hierarchical model:

The mathematical formulation of the model is:

$$\begin{aligned}\mu &\sim N(180, 22) \\ \tau &\sim \text{exponential}(0.02) \\ \mu_d &\sim N(\mu, \tau) \\ \sigma_d &\sim \text{exponential}(0.02) \\ w_{i,d} &\sim N(\mu_d, \sigma_d)\end{aligned}$$

Unlike other models, in a hierarchical model, a multilevel modeling process is considered, assuming that the prior distributions depend on an upon further hyperparameters that are obtained from a hyperprior distributions.

The hierarchical model, which eliminates the assumption that only a single set of data represents the entire population by employing a multilevel approach, yields more realistic results compared to the separate model, where the standard deviation and mean are considered dependent on each group individually. On the other hand, the pooled model also outperforms the separate model, though it falls short of the hierarchical model in delivering accurate results.

2.16 (I)

- Separate model:

```
data {
  int<lower=0> N_observations;
  int<lower=0> N_diets;
  array[N_observations] int diet_idx; // Pair observations to their diets.
  vector[N_observations] weight;
}

parameters {
  vector[N_diets] mean_diet;
  vector<lower=0>[N_diets] sd_diet;
}

model {
  for (diet in 1:N_diets) {
    mean_diet[diet] ~ normal(180, 22);
    sd_diet[diet] ~ exponential(0.02);
  }

  for (obs in 1:N_observations) {
    weight[obs] ~ normal(mean_diet[diet_idx[obs]], sd_diet[diet_idx[obs]]);
  }
}

generated quantities {
  real weight_pred = normal_rng(mean_diet[4], sd_diet[4]);
  real mean_five = normal_rng(180, 22);
}
```

- Pooled model:

```
data {  
  int<lower=0> N_observations;  
  int<lower=0> N_diets;  
  array[N_observations] int diet_idx; // Pair observations to their diets.  
  vector[N_observations] weight;  
}  
  
parameters {  
  real mean_diet;  
  real<lower=0> sd_diet;  
}  
  
model {  
  mean_diet ~ normal(180, 22);  
  sd_diet ~ exponential(0.02);  
  
  for (obs in 1:N_observations) {  
    weight[obs] ~ normal(mean_diet, sd_diet);  
  }  
}  
  
generated quantities {  
  real weight_pred = normal_rng(mean_diet, sd_diet);  
  real mean_five = normal_rng(180, 22);  
}
```

- Hierarchical model:

```
data {
  int<lower=0> N_observations;
  int<lower=0> N_diets;
  array[N_observations] int diet_idx; // Pair observations to their diets.
  vector[N_observations] weight;
}

parameters {
  real hyper_mu;
  real<lower=0> tau;
  vector[N_diets] mean_diet;
  vector<lower=0>[N_diets] sd_diet;
}

model {
  hyper_mu ~ normal(180, 22);
  tau ~ exponential(0.02);

  for (diet in 1:N_diets) {
    mean_diet[diet] ~ normal(hyper_mu, tau);
    sd_diet[diet] ~ exponential(0.02);
  }

  for (obs in 1:N_observations) {
    weight[obs] ~ normal(mean_diet[diet_idx[obs]], sd_diet[diet_idx[obs]]);
  }
}

generated quantities {
  real sd_diets = sd_diet[4];
  real weight_pred = normal_rng(mean_diet[4], sd_diet[4]);
  real mean_five = normal_rng(hyper_mu, tau);
}
```

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

2.17 (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"
)
```

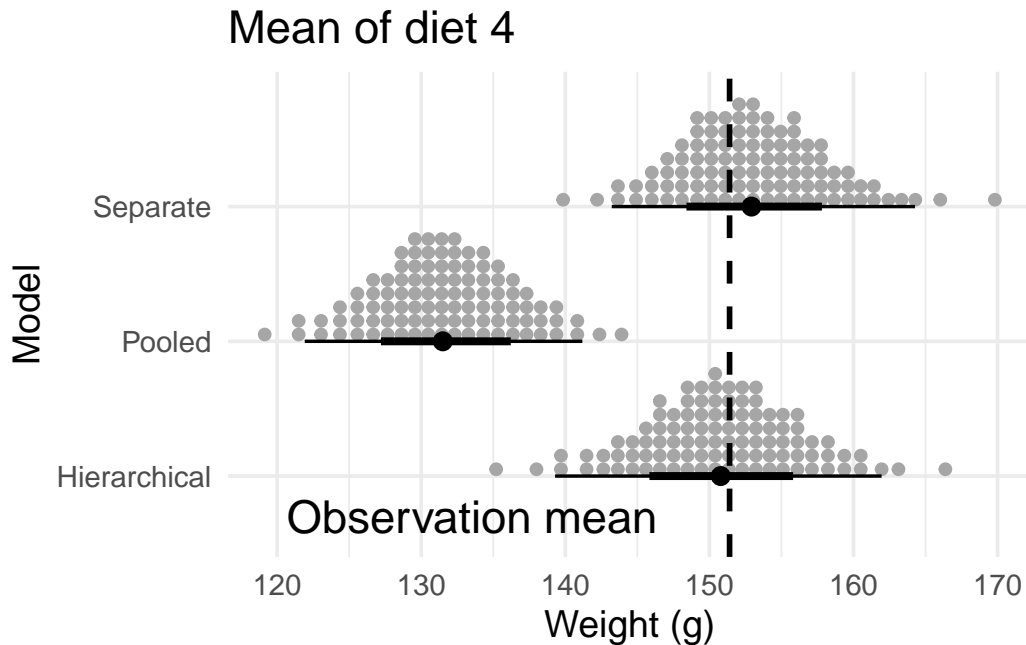


Figure 1: Posterior distribution of the mean weight of chicks consuming diet 4.

1. In the Separate Model, the data points exhibit a wide dispersion of weights, reflecting a substantial variability in the predicted outcomes. This implies that the Separate model permits a high degree of customization for each diet, potentially capturing a broad spectrum of effects, albeit at the risk of overfitting the data.
2. In contrast, the Pooled Model displays data points tightly clustered around a specific weight, indicative of limited variability in the predicted outcomes. This hints at the Pooled model's assumption that all diets have identical effects, which can be a robust simplification but may miss individual diet-specific nuances.
3. The Hierarchical Model strikes a balance between the Separate and Pooled models. It presents data points with moderate dispersion, suggesting a moderate degree of variability in the predicted weights. This indicates that the Hierarchical model combines the advantages of both individual diet customization and shared information between diets, achieving a nuanced approach to the problem.

2.18 (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"
  )
```

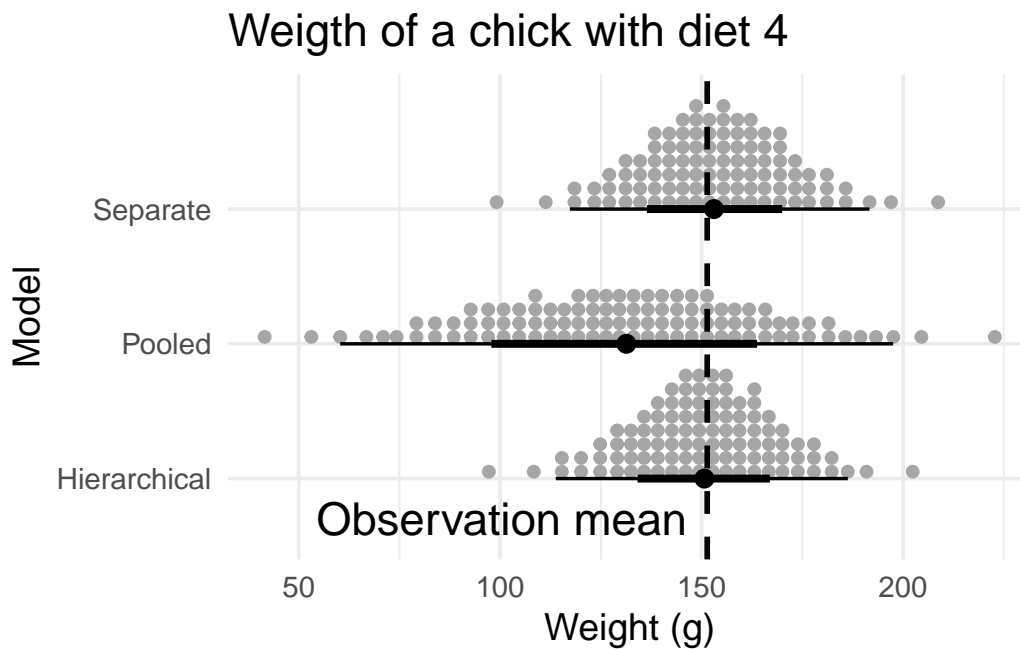


Figure 2: The (posterior) predictive distribution of the weigh of a chick consuming diet 4.

1. Separate Model: This model exhibits a notably higher average weight prediction for chicks on diet 4 and a narrower weight range compared to the other two models. In essence, it implies that the separate model offers a more specific weight prediction for this diet, indicating limited variability between different diets.
2. Pooled Model: In the pooled model, all groups are treated as if they share a common distribution, which means the spread in predictions is influenced by observations from all groups. This approach may lead to a broader overall spread, especially if there is inherent variability between groups.
3. Hierarchical Model: Positioned between the separate and pooled models, the hierarchical model assumes that group parameters stem from a shared distribution. This setup allows for some between-group variability while also facilitating information sharing between groups. As a result, it may produce a slightly broader spread than the separate model if between-group variation exists but less than what's observed in the pooled model. In essence, the hierarchical model strikes a balance between the extremes of the separate and pooled models.

2.19 (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")
```

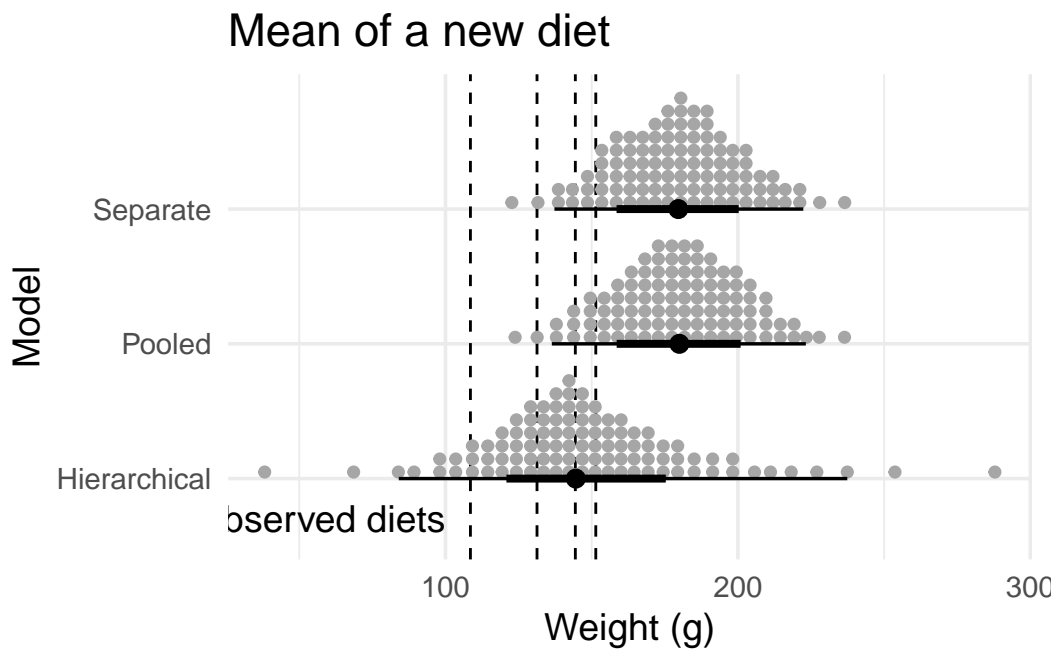


Figure 3: Posterior distribution of the mean weight of chicks consuming the new diet 5 not seen before.

1. Separate Model: In the Separate model, the posterior distribution is characterized by a moderate level of concentration, falling between the Pooled and Hierarchical models. This implies that the Separate model yields a range of credible values for the mean weight of the new diet that is neither too narrow nor too broad.
2. Pooled Model: The Pooled model's posterior distribution is tightly centered around a specific value, signifying a low degree of uncertainty. This outcome suggests that the Pooled model, which assumes uniform effects across all diets, produces a constrained set of credible values for the mean weight of the new diet.
3. Hierarchical Model: Under the Hierarchical model, the posterior distribution for the mean weight of the new fifth diet exhibits a wide dispersion, indicating a high level of uncertainty. This implies that the Hierarchical model results in a broad range of credible values for the mean weight of the new diet, offering a less precise estimate compared to the other models.

2.20 (p)

```
posterior_expectation_separate <- mean(mean_diet_4_separate)
posterior_expectation_pooled <- mean(mean_diet_4_pooled)
posterior_expectation_hierarchical <- mean(mean_diet_4_hierarchical)

credible_interval_separate <- quantile(mean_diet_4_separate, c(0.05, 0.95))
credible_interval_pooled <- quantile(mean_diet_4_pooled, c(0.05, 0.95))
credible_interval_hierarchical <- quantile(mean_diet_4_hierarchical, c(0.05, 0.95))

cat("Separate Model:\nPosterior Expectation =", round(posterior_expectation_separate,2) ,
    "\n90% Credible Interval =", round(credible_interval_separate,2), "\n\n")
```

Separate Model:

Posterior Expectation = 153.16

90% Credible Interval = 144.87 162.11

```
cat("Pooled Model:\nPosterior Expectation =", round(posterior_expectation_pooled,2),
    "\n90% Credible Interval =", round(credible_interval_pooled,2), "\n\n")
```

Pooled Model:

Posterior Expectation = 131.63

90% Credible Interval = 123.67 139.55

```
cat("Hierarchical Model:\nPosterior Expectation =", round(posterior_expectation_hierarchical,2),
    "\n90% Credible Interval =", round(credible_interval_hierarchical,2), "\n\n")
```

Hierarchical Model:

Posterior Expectation = 150.77

90% Credible Interval = 141.56 159.88

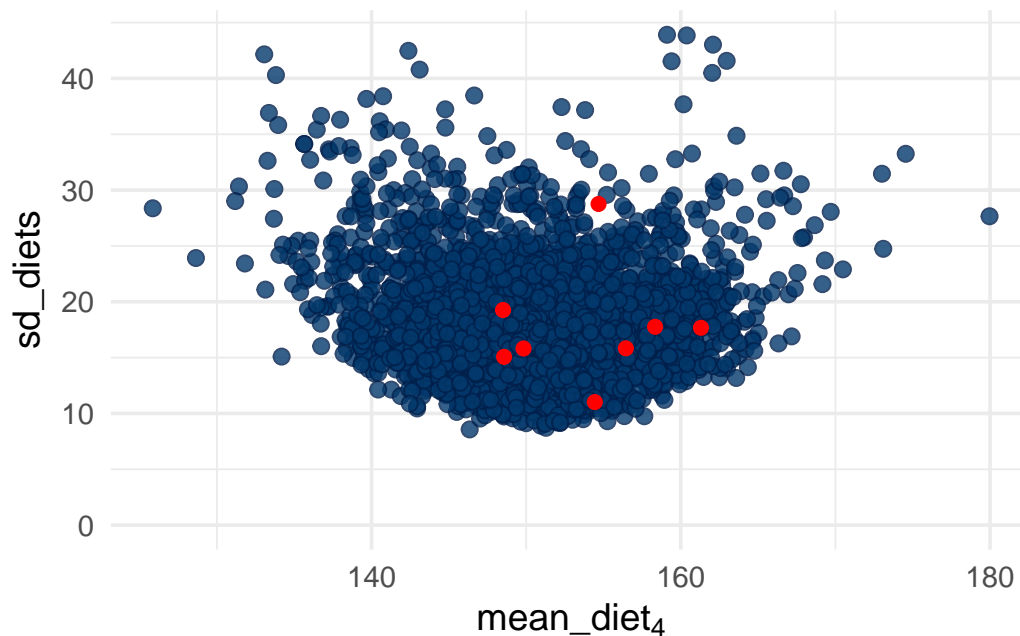
3 Hierarchical model with BRMS (3p)

3.1 (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.



The scatter plot reveals red points scattered amidst the blue ones, indicating divergent transitions in the Stan Hamiltonian Monte Carlo sampling process. These divergent transitions often signal potential issues with the model fit, particularly when the model is overly intricate or the priors lack robust constraints, leading the sampler into extreme parameter territories.

3.2 (b)

Because `brms` is a bit chatty, suppress its 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 code you used to create the brms model and run the sampling
brms_fit = brm(
  weight ~ 1 + (1 | Diet),
  data=Chick12,
  prior=c(
    # REPLACE WITH YOUR PRIOR FOR THE INTERCEPT
    prior(normal(180, 22), class="Intercept"), # prior for mu_0
    # REPLACE WITH YOUR PRIOR FOR SIGMA
    prior(exponential(0.02), class="sigma"),      # prior for sigma
    # REPLACE WITH YOUR PRIOR FOR SD
    prior(exponential(0.02), class="sd")          # prior for tau
  ),
  backend = "cmdstanr",
  save_pars = save_pars(manual = c("z_1[1,4]"))
)
```

3.3 (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.
# Calculate the 90% credible interval for mu_4
credible_interval <- quantile(mu_4, probs = c(0.05, 0.95))

# Calculate the posterior mean for mu_4
posterior_mean <- mean(mu_4)

# Display the results
cat("Posterior Mean for mu_4:", round(posterior_mean, 2), "\n")
```

Posterior Mean for mu_4: 149.65

```
cat("90% Credible Interval for mu_4:", round(credible_interval[1], 2), "to", round(credible_interval[2], 2), "\n")
```

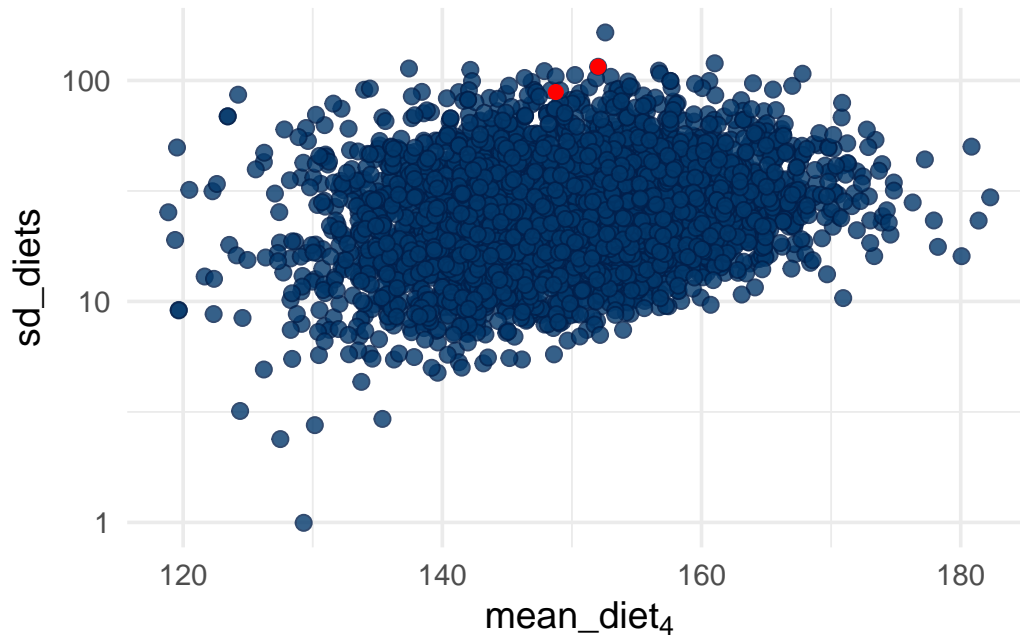
90% Credible Interval for mu_4: 134.9 to 164.48

3.4 (d)

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))
```



Certainly, the preceding plot illustrates a reduced presence of red points amid the blue ones, signifying fewer instances of divergent transitions in the Hamiltonian Monte Carlo (HMC) sampling process utilized by Stan. This observation suggests that the HMC sampling process is achieving good convergence. With fewer divergent transitions, the sampler effectively explores the posterior distribution, resulting in more dependable and precise parameter estimates.

3.5 (e)

- The non-centered version, run using brms, exhibits a lower occurrence of divergences. Moreover, the second parameterizations resulted in fewer divergent transitions due to the fewer number of red points.
- In the case of the non-centered version, most divergences are characterized by high values of the tuning parameter τ around 100. However, in the centered version, no distinct pattern emerges; instead, roughly 75% of the divergences occur at low τ values, typically less than 20.
- Interestingly, the centered version displays issues that are less localized and appear to be distributed across various regions, lacking a specific concentration.