

Assignment 7

anonymous

General information

AI was used in handling the stan code of the three models.

Hierarchical Model: Chicken Data with Stan (6p)

Choosing a weakly informative prior by intuition

2.(a)

In my opinion a typical weight range for a fully grown chicken is probably between 1500 grams to 3000 grams. Because I have searched some related information online as there are many people said broiler chickens, which are raised for meat, often weigh more, reaching the higher end of this range, while egg-producing chickens tend to be on the lighter side. In general consensus among people, no matter what types of chickens they are, their weights are almost from 1500 grams to 3000 gram when they fully grow up.

2.(b)

For a 12-day old chick, considering that a chick starts from a really small grams at birth and hasn't undergone major growth spurts by day 12, By searching some online information I'd adjust the range to approximately 80 grams to 200 grams. Therefore, a reasonable choice for the mean μ_0 would be the midpoint of this range, which is 140 grams. I think this estimation considers the chick's modest growth from birth and positions it at a stage before it undergoes more rapid weight gain as it approaches adulthood.

2.(c)

Because we select the range of 80 grams to 200 grams, so a conservative approach would be to slightly expand this range to account for small or large chicks. Let's set the lower bound at 60 grams and the upper bound at 220 grams. This expanded range ensures that we capture most of chicks' data, taking into account factors like breed variations, health conditions, or differences in nutrition and care that might lead to weights outside the average expected range.

2.(d)

Based on the above range 60 grams to 220 grams, one way to think about it is that this range covers about 4 standard deviations, as we have chosen a fairly conservative range. So $(220-60) / 4$ is equal to 40 grams. Thus, a possible standard deviation of 12-day-old chick weight is 40 grams. plausible standard deviation = 40 grams

According to the tips from question, the weakly informative standard deviation is 10 times the possible standard deviation of the data. Based on the 40 grams above, this means that we should choose a standard deviation σ of 400 grams as the standard deviation of the prior. This ensures that our priors are loose and do not overly restrict the formation of a posterior.

Under the above recommendation, the standard deviation = 400 grams

Using the expression $\mu \pm 2\sigma$: Lower bound: $140 - 2(400) = -660$ grams Upper bound: $140 + 2(400) = 940$ grams

A chick cannot have a negative weight, though it emphasizes the weakly informative nature of the prior. It suggests we are very uncertain about our prior beliefs, and the data will strongly influence the posterior. The primary purpose is to be very broad and not to overly influence the resulting model once actual data is applied. ## 2.(e) Given our choices, the final prior for the mean weight μ of 12 days-old chicks is defined as:

$$\mu \sim N(140, 400)$$

Where $\mu_0 = 140$ grams, the standard deviation $\sigma_0 = 400$ grams.

Choosing a weakly informative prior using external references

2.(f)

According to The Humane League(2021),one-week-old broilers weigh between 45 and 60 grams while two-week-old broilers weigh between 90 and 170 grams Therefore, we can infer that the weight of a 12-day-old chick is probably somewhere in between these two ranges, so we just assume the weight range for 12-day old chicken is probably between 80 grams to 160 grams.

Reference: The Humane League. (2021, April 1). The Average Chicken Weighs a Lot More Than They Used To. Retrieved October 26, 2023, from <https://thehumaneleague.org/article/average-chicken-weight>

2.(g)

Mean of our weakly informed prior: 120 grams

Because the weight range of 80 to 160 grams for a 12-day-old chick, a simple approach to choosing a mean for the weakly informed prior would be to take the

arithmetic mean of these two extremes, which yields 120 grams. This choice makes an assumption of symmetry, which suggests that it is equally likely for the chicks to fall above or below this mean within the given range. This method of setting the mean provides a central value within the specified range, making no further assumptions about the distribution of weights within this range, which aligns with the idea of having a weakly informed prior.

2.(h)

Because the range is 80 grams and 160 grams for a 12-day-old chick.

Using the upper bound $b=160$ grams: $\mu+3\sigma<160$.

Substituting the value of $\mu=120$ grams: $120+3\sigma<160$ Now solve for $3\sigma<160-120$, $3\sigma<40$ $\sigma\approx 13.33$.

Using the lower bound $b=80$ grams: $\mu-3\sigma>80$

Substituting the value of $\mu=120$ grams: $120-3\sigma>80$

Now solve for $3\sigma>120-80$, $3\sigma>40$, $\sigma\approx 13.33$

Both bounds give the same result for σ , which is approximately 13.33 grams.

2.(i)

Based on our calculation, the final prior for the mean weight μ of 12 days-old chicks is defined as:

$$\mu \sim N(120, 13.33)$$

Where $\mu = 120$ grams, the standard deviation $\sigma = 13.33$ grams.

2.(j)

Case1: The weight of the chicken is a non-negative variable, whereas the normal distribution is defined over the entire real number line, including negative values. This means that the normal distribution will assign some negative probability to the weight of the chick, which does not make sense in reality. Therefore, in this case, the normal distribution may not be the best choice. A more appropriate choice might be a distribution defined only on non-negative values, such as a gamma distribution.

Case2: Although the weight of a chick is not strictly bounded, in some cases there may be a practical upper limit. For example, a chick may not weigh more than a certain value due to biological limitations. The normal distribution does not naturally truncate at a certain point, but continues to assign probabilities to weights beyond that point. In this case, it may be necessary to consider a bounded distribution, such as a beta distribution.

Modeling diet effects on chicken weight

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

```
data("ChickWeight")

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

## [1] 129.2449

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
```

Sample from the posterior

2.(k)

Separate Model:

$$\mu_d \sim \pi(\mu_d)$$

This represents the mean weight of the chick for a specific diet d.

$$\sigma_d \sim \pi(\sigma_d)$$

This is the standard deviation for the weight of the chick for a specific diet d,

$$w_{i,d} \sim N(\mu_d, \sigma_d)$$

This gives the weight of each individual chick i on diet d. In the separate model, each diet is modeled individually. Every diet has its own unique parameters for the mean and standard deviation of chick weights. This model does not share information between different diets.

Pooled Model:

$$\mu_d \sim \pi(\mu_d) = N(\mu_0, \sigma_0)$$

This is the mean weight of the chick across all diets, where μ and σ are hyperparameters. In the pooled model, measurements from all the diets are combined, and there's no distinction between them. Essentially, it assumes that all

diets have the same effect on chick weights, and it estimates a single overall mean and standard deviation.

Hierarchical Model:

$$\mu \sim N(\mu, \tau)$$

$$\mu \sim \pi(\mu) = N(\mu_0, \sigma_0)$$

This is the mean of the prior for diet-wise mean weights. The hierarchical model introduces an additional layer of hierarchy. The diet-wise mean weights still have their individual distributions (like in the separate model), but the parameters of these distributions are themselves drawn from shared prior distributions. This model essentially borrows strength from all diets to inform the estimates for each individual diet. It lies in between the separate and pooled models in terms of structure.

Difference between the Models: Separate Model: Assumes each diet is unique and does not share information between diets. Pooled Model: Combines all diets and treats them as if they are the same, estimating a single set of parameters for all of them. Hierarchical Model: Sits between the separate and pooled models. It assumes each diet has its own parameters, but these parameters are drawn from a common distribution, thus sharing information across diets.

Generally, while the separate model treats each diet as its own entity and the pooled model assumes a common effect across all diets, the hierarchical model introduces a compromise by assuming shared underlying distributions for the parameters of each diet.

2.(I)

The stan code for separate model is below:

```
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 {  
  // Average weight of chicks with a given diet.  
  vector[N_diets] mean_diet;  
  
  // Standard deviation of weights observed among chicks sharing a diet.  
  vector<lower=0>[N_diets] sd_diet;  
  real<lower=0> hyper_mean_diet;  
  real<lower=0> hyper_sd_diet;  
}
```

```

model {
  // Priors
  // These look bad. I need to think about these again.
  hyper_mean_diet ~ normal(120, 13.33);
  hyper_sd_diet ~ exponential(.02);
  for (diet in 1:N_diets) {
    mean_diet[diet] ~ normal(120, 13.33);
    sd_diet[diet] ~ exponential(.02);
  }

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

  // Best practice would be to write the likelihood without the for loop
  // as:
  // weight ~ normal(mean_diet[diet_idx], sd_diet[diet_idx]);
}

generated quantities {
  real weight_pred;
  real mean_five;
  // The below is just there to make the plotting in the template work
  // with the "wrong model".
  real sd_diets = sd_diet[4];

  // Sample from the (posterior) predictive distribution of the fourth
  // diet.
  weight_pred = normal_rng(mean_diet[4], sd_diet[4]);
  // Construct samples of the mean of the fifth diet.
  // We only have the prior...
  mean_five = normal_rng(hyper_mean_diet, hyper_sd_diet);
}

```

The stan code for pooled model is below:

```

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 {
  // Priors
  mean_diet ~ normal(120, 13.33);
  sd_diet ~ exponential(0.02);

  // Likelihood
  weight ~ normal(mean_diet, sd_diet);
}

generated quantities {
  real weight_pred;
  real mean_five;
  weight_pred = normal_rng(mean_diet, sd_diet);
  mean_five = mean_diet;
}

```

The stan code for hierarchical model is below:

```

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 mu; // Mean of prior for diet-wise mean weights
  real<lower=0> tau; // Standard deviation of prior for diet-wise mean weights
  vector[N_diets] mean_diet; // Diet-wise mean weights
  vector<lower=0>[N_diets] sd_diet; // Diet-wise standard deviations
}

model {
  // Priors
  mu ~ normal(120, 13.33); // Prior for the mean of diet-wise mean weights
  tau ~ exponential(0.02); // Prior for the standard deviation of diet-wise mean weights
  mean_diet ~ normal(mu, tau); // Diet-wise mean weights
  sd_diet ~ exponential(0.02); // Diet-wise standard deviations

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

```

```

}

generated quantities {
  real weight_pred;
  real mean_five;
  real sd_diets = sd_diet[4];
  real new_diet_mean;
  real sigma;
  sigma = mean(sd_diet);
  new_diet_mean = normal_rng(mu, tau);
  weight_pred = normal_rng(mean_diet[4], sd_diet[4]);
  mean_five = normal_rng(new_diet_mean, sigma);
}

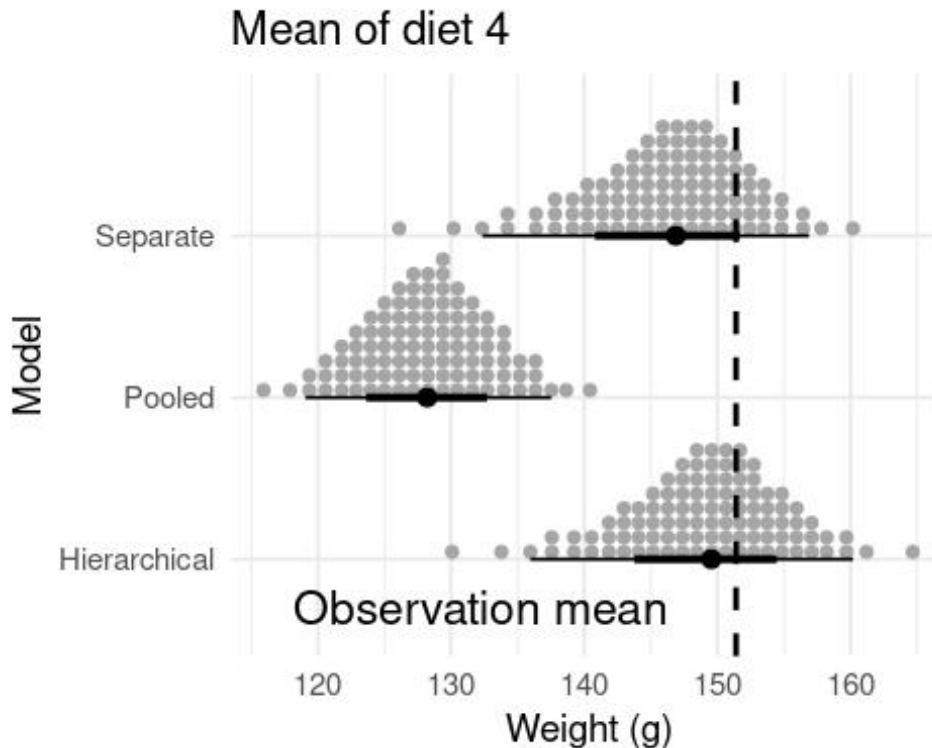
```

2.(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"
  )

```

Separate model: The data points in this model are concentrated and show independent estimates of the weight of chicks on each diet. As can be seen from the figure, the estimates of the Separate model have certain variability and cover a certain weight range. Besides the center of the Separate model is the second closest to the observation mean among the three models.

Pooled model: The Pooled model appears to have a narrow distribution, with data from all diets being pooled together for estimates. As can be seen from the figure, the Pooled model has a more concentrated posterior distribution, indicating that the model has a more certain estimate of the weight of chicks on the fourth diet.

Hierarchical model: Hierarchical models have data point distributions between Separate and Pooled models. This is because Hierarchical models take into account the correlations between diets, but also provide some flexibility for each diet. As can be seen from the figure, the Hierarchical model has a slightly more concentrated posterior distribution than the Separate model, but is slightly more dispersed than the Pooled model.

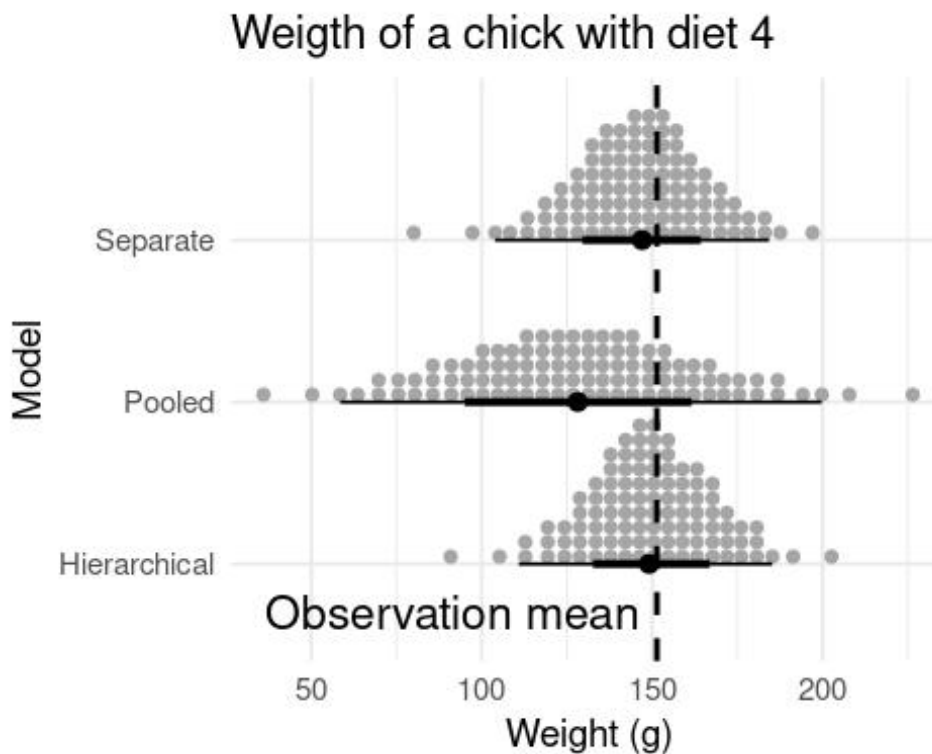
2.(n)

```
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"
)

```



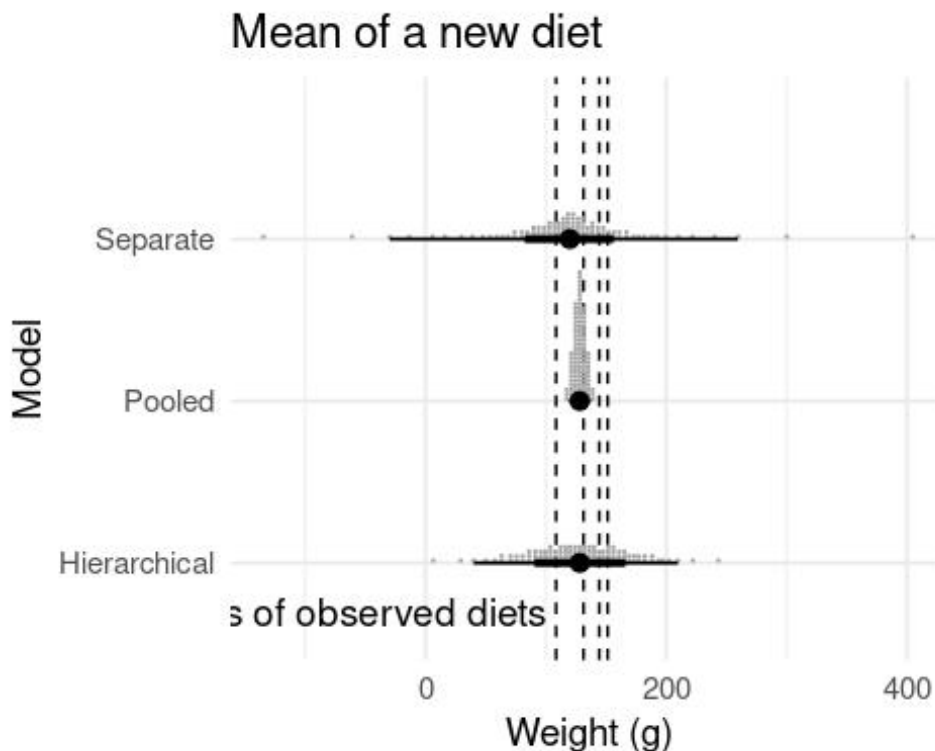
Separate model:In the Separate model, the point distribution indicates that the predicted weight distribution for feed 4 is relatively narrow but not the narrowest, meaning that there is some certainty about the predicted weight for this particular feed. The Separate model estimates parameters separately for each feed, so its predictions are based only on feed 4 data. The observed average weight is close to the center of the model's predictions, indicating that the Separate model's predictions are generally consistent with the observed data.

Pooled model:In the Pooled model, the distribution of points is concentrated over a wide range, indicating that the model has a high degree of uncertainty for weight predictions across all feeds. The results were not very good, and it is the worst of the three models.

Hierarchical model:In the Hierarchical model, the distribution of points is the most concentrated of the three models. This model takes into account both the uniqueness of each feed and the shadow of the overall distribution. The prediction turned out to be good relatively.

2.(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")
```



Separate Model: The posterior distribution for the separate model shows a very wide range of possible mean weights. This suggests high uncertainty, which could be due to having less data for each diet or not borrowing strength from the other diets' data. The model treats each diet as completely separate from the others.

Pooled Model: The pooled model's posterior distribution is concentrated around a specific value with very little spread, indicating a high degree of certainty about the

mean weight. This is because the pooled model assumes that all diets have the same mean effect, and it uses all the data to estimate this common mean.

Hierarchical Model: The hierarchical model offers a compromise between the separate and pooled models. It allows for variation between the diets but also shares information among them. The posterior distribution for the hierarchical model is narrower than for the separate model but wider than for the pooled model, indicating a balanced amount of uncertainty that takes into account both the variability among diets and the common information shared across them.

2.(p)

Hierarchical model with BRMS (3p)

Assuming you have fit your models and named them fit_separate, fit_pooled, and fit_hierarchical

For the Separate model:

```
samples_separate <- extract_variable(fit_separate, "mean_diet[4]")
posterior_expectation_separate <- mean(samples_separate)
lower_bound_separate <- quantile(samples_separate, 0.05)
upper_bound_separate <- quantile(samples_separate, 0.95)
```

For the Pooled model:

```
samples_pooled <- extract_variable(fit_pooled, "mean_diet")
posterior_expectation_pooled <- mean(samples_pooled)
lower_bound_pooled <- quantile(samples_pooled, 0.05)
upper_bound_pooled <- quantile(samples_pooled, 0.95)
```

For the Hierarchical model:

```
samples_hierarchical <- extract_variable(fit_hierarchical, "mean_diet[4]")
posterior_expectation_hierarchical <- mean(samples_hierarchical)
lower_bound_hierarchical <- quantile(samples_hierarchical, 0.05)
upper_bound_hierarchical <- quantile(samples_hierarchical, 0.95)
```

```
print(paste("Separate Model: Posterior Expectation =", round(posterior_expectation_separate, 2),
            "90% CI =", round(lower_bound_separate, 2), "-", round(upper_bound_separate, 2)))
```

```
## [1] "Separate Model: Posterior Expectation = 146.26 90% CI = 135.48 - 155.03"
```

```
print(paste("Pooled Model: Posterior Expectation =", round(posterior_expectation_pooled, 2),
            "90% CI =", round(lower_bound_pooled, 2), "-", round(upper_bound_pooled, 2)))
```

```
## [1] "Pooled Model: Posterior Expectation = 128.18 90% CI = 120.43 - 136.07"

print(paste("Hierarchical Model: Posterior Expectation =", round(posterior_expectation_hierarchical, 2),
            "90% CI =", round(lower_bound_hierarchical, 2), "-", round(upper_bound_hierarchical, 2)))

## [1] "Hierarchical Model: Posterior Expectation = 149.14 90% CI = 138.25 - 158.29"
```

Separate Model: Posterior Expectation = 146.26 90% CI = 135.48 - 155.03

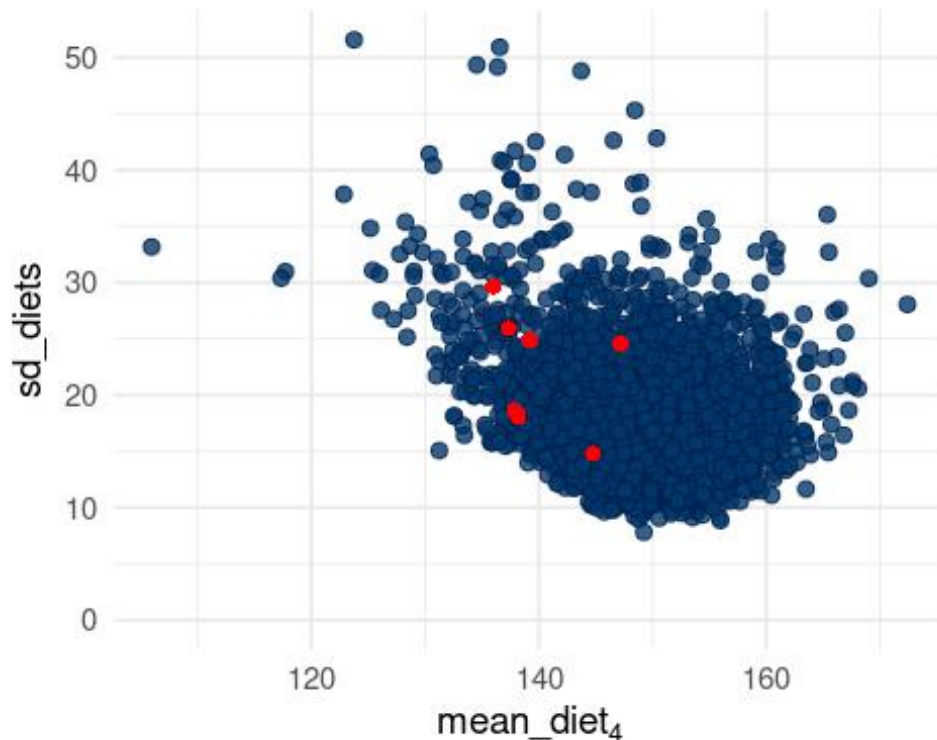
Pooled Model: Posterior Expectation = 128.18 90% CI = 120.43 - 136.07

Hierarchical Model: Posterior Expectation = 149.14 90% CI = 138.25 - 158.29

3.(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 estimated average body weight of chickens on the fourth diet appears to be concentrated in the range of 140 to 155, with some uncertainty. The weight variability of chickens on the fourth diet was estimated to be highest between 15 and 25.

3.(b)

Create a brms model and sample from the posterior

I used the weekly informative prior in 2.a), which is $N(120, 13.33)$.

```
brms_fit = brm(
  weight ~ 1 + (1 | Diet),
  data=Chick12,
  prior=c(
    # REPLACE WITH YOUR PRIOR DERIVED in 2)
    prior(normal(120,13.33), class="Intercept"),
    prior(exponential(.02), class="sd"),
    prior(exponential(.02), class="sigma")
  ),
  #backend = "cmdstanr",
  save_pars = save_pars(manual = c("z_1[1,4]"))
)
```

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.

```
mean_mu_4 <- mean(mu_4)
```

```
q5_mu_4 <- quantile(mu_4, probs = 0.05)
```

```
q95_mu_4 <- quantile(mu_4, probs = 0.95)
```

```
cat("Mean of mu_4:", round(mean_mu_4, 2), "\n")
```

```
## Mean of mu_4: 146.24
```

```
cat("5% quantile of mu_4:", round(q5_mu_4, 2), "\n")
```

```
## 5% quantile of mu_4: 130.35
```

```
cat("95% quantile of mu_4:", round(q95_mu_4, 2), "\n")
```

```
## 95% quantile of mu_4: 161.51
```

```
cat("90% CI =", round(q5_mu_4, 2), "-", round(q95_mu_4, 2))
```

```
## 90% CI = 130.35 - 161.51
```

```
Mean of mu_4: 146.24
```

```
90% CI = 130.35 - 161.51
```

3.(d)

```
draws = as_draws_df(brms_fit) |>
```

```
  posterior::mutate_variables(mean_diet_4 = `r_Diet[4,Intercept]` + b_I  
  ntercept)
```

```
bayesplot::mcmc_scatter(draws,
```

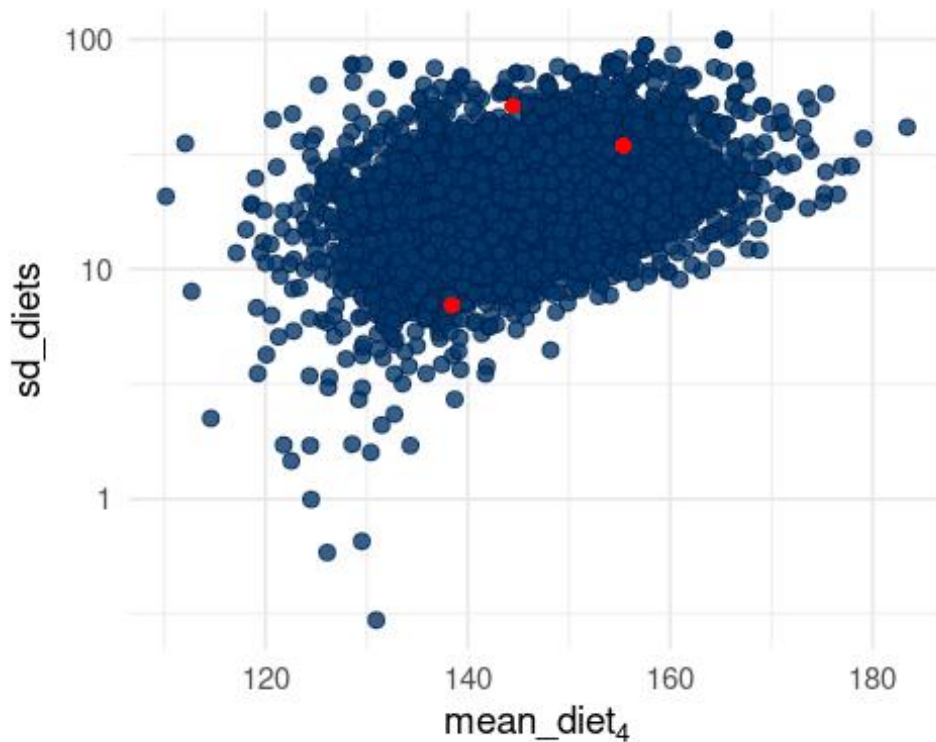
```
  pars = c("mean_diet_4", "sd_Diet__Intercept"),
```

```
  np = nuts_params(brms_fit)) +
```

```
  scale_y_log10() +
```

```
  xlab(expression(mean_diet[4])) +
```

```
  ylab(expression(sd_diets))
```



Most data points cluster between 140 and 170 for mean_diet4, suggesting this is the typical range. There's a broad horizontal pattern, indicating that within the primary range, changes in mean_diet4 don't greatly influence sd_diets. There are potential outliers around values 120 and 180 for mean_diet4, where sd_diets values are unusually low. The plot uses a logarithmic scale for the sd_diets axis, emphasizing the difference between its low and high values. The deeper blue regions indicate areas of high data density, primarily where sd_diets lies between 10 and 60. The incidence of red divergence points has reduced by approximately 50%. Implementing the new model has contributed to a notable enhancement in divergence outcomes.

3. (e)

The utilization of non-centered parameterizations led to a reduction in divergent transitions.

In the model with centered parameterizations, divergences tend to group around where τ is close to 0. On the other hand, in the model with non-centered parameterizations, the divergences appear to be sparse with no discernible pattern.

Indeed, the centered parameterizations model encounters issues when sampling from areas with elevated τ values, as evidenced by the sparser distribution of points in that region from the plot.