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

2.2 (a)

Based on my own past experience a fully grown chicken in grams is around 1-3 kilograms. This is based that mostly on the price of a chicken in the groaceary store where the kilo price is around 10-15 euros and a full chicken is about 20-30 euros. Although this the chicken as food has removed legs, head, skin and so on which remove some weigth. The assumption has taken this into account.

2.3 (b)

A 12-day old chicken range is probably around 50 to 300 grams with a mean at 200 grams. This is based on that a litle chiken fits into my pawn and other small animals as kittens which are in the same size weigh a few hounded grams.

2.4 (c)

A conservative lower and upper bound for the weight of any 12-day old chick 50 grams to 500 grams (mean =225). This range is wider to try to eliminate impossible values. It is impossible for a n chicken to weigh less than egg therefore we chose the weight of an egg as an lower bound. This excludes the notion that the chiken grew something in the first 12 days. Well eggs can get quite big for example a ostrich egg so an egg does probably not have 10x difference between lowest and highest possible value. Therefore we chose 500 grams.

2.5 (d)

Plausible Standard deviation is probably around 35 in the primary estimation and around 100 in the conservative estimation. Both of the estimation ranges are outside the (mean -+ 3 x standard deviation)

2.6 (e)

$$\mu \sim N(200, 35^2)$$

2.7 Choosing a weakly informative prior using external references

2.8 (f)

This is based on a interview on a family friend farmer. A typical weight range for a fully grown commercial broiler chicken might be between 2 to 4 kilograms depending on the breed of a chicken. It takes around 5-6 weeks for the chicken to reach adulthood. An broiler chick typically weighs around 50 grams. This means with a constant growth rate they grow around 300 grams a week. this means a chicken is around 500 grams at 12 days. This ranges is also in line with sources https://en.wikipedia.org/wiki/Broiler. Where for some breeds it takes 30 days for a chicken to get to 1.5 kg.

2.9 (g)

In light of the earlier conclusions a mean of 500 grams would be optimal mean with an range from 200 to 800 as my earlier predictions where way off. To make sure the true mean and variablity is the range.

2.10 (h)

$$\sigma\_0 = \frac{b_0 - \mu_0}{3} = \frac{800 - 500}{3} = 100$$

2.11 (i)

$$\mu \sim N(500, 100^2)$$

2.12 (j)

Example cases include variables that are heavy tailed, meaning that there are more extreme values than would be expected under a normal distribution.

2.13 Modeling diet effects on chicken weight

a separate model a pooled model a hierarchical model

! 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
```

2.14 (k) Math

2.15 Default assumptions between all models

There are three models Separate model, Pooled model and Hierarchical model. All of these models try to predict the weight of the chicks at day 12. They use different approaches to predict. Firstly, The separate model assumes that each diet is modeled individually. The reason for using this model is to assume there is no pattern between diets. Secondly, the Pooled model or jointly assumes that the diet has no effect on the weight to predictions. It is trivial that neither case is true. Therefore the third model (Hierarchical) tryes to combine both approches.

For the separate model there will be separated weights $w_i = N(\mu_i, \sigma_i)$, where prior μ_i is presented earlier in the assignment and prior σ_i is just $\exp(0.02)$.

For the Pooled model there is only on set of weights. $w = N(\mu, \sigma)$, where μ_i is prior presented earlier in the assignment and σ is just $\exp(0.02)$.

For the Hierarchical model there is an $w_i = N(\mu_{i,d}, \sigma)$, where $\mu_{i,d} \sim N(\mu, \sigma)$ and σ is just $\exp(0.02)$. Therefore this takes an independent mean for each but uses the same standard deviation.

2.16 (l) Hierarchical model

2.17 (l) Separate model

2.18 (l) Pooled model

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.

```
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;
  real mean_prior;
  real sigma_prior;
}

parameters {
  real mu_0;
  vector<lower=0>[N_diets] sigma_0;

  vector[N_diets] mean_diet;
  real<lower=0> sd_diet;

}

model {
  mu_0 ~ normal(mean_prior, sigma_prior);
  sigma_0 ~  exponential(0.02);

  mean_diet ~ normal(mu_0, sigma_0);
  sd_diet ~ exponential(0.02);

  weight ~ normal(mean_diet[diet_idx], sd_diet);
}

generated quantities {
  real weight_pred;
  real mean_five;
  real sd_diets = sd_diet;

  weight_pred = normal_rng(mean_diet[4], sd_diet);

  mean_five = normal_rng(mu_0, sd_diet);
}
```

Figure 1: 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;
  real mean_prior;
  real sigma_prior;
}

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;
}

model {
  // Priors
  for (diet in 1:N_diets) {
    mean_diet[diet] ~ normal(mean_prior, sigma_prior);
    sd_diet[diet] ~ exponential(.02);
  }

  weight ~ normal(mean_diet[diet_idx], sd_diet[diet_idx]);
}

generated quantities {
  real weight_pred;
  real mean_five;
  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(mean_prior, sigma_prior);
}
```

Figure 2: Separate model

```
data {
  int<lower=0> N_observations;
  vector[N_observations] weight;
  real mean_prior;
  real sigma_prior;
}

parameters {
  // Average weight of chicks with a given diet.
  real mean_diet;
  // Standard deviation of weights observed among chicks sharing a diet.
  real<lower=0> sd_diet;
}

model {
  mean_diet ~ normal(mean_prior, sigma_prior);
  sd_diet ~ exponential(.02);

  weight ~ normal(mean_diet, sd_diet);
}

generated quantities {

  real weight_pred;
  real mean_five;

  weight_pred = normal_rng(mean_diet, sd_diet);
  mean_five   = normal_rng(mean_diet, sd_diet);
}
```

Figure 3: Pooled model

2.19 (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"
  )
```

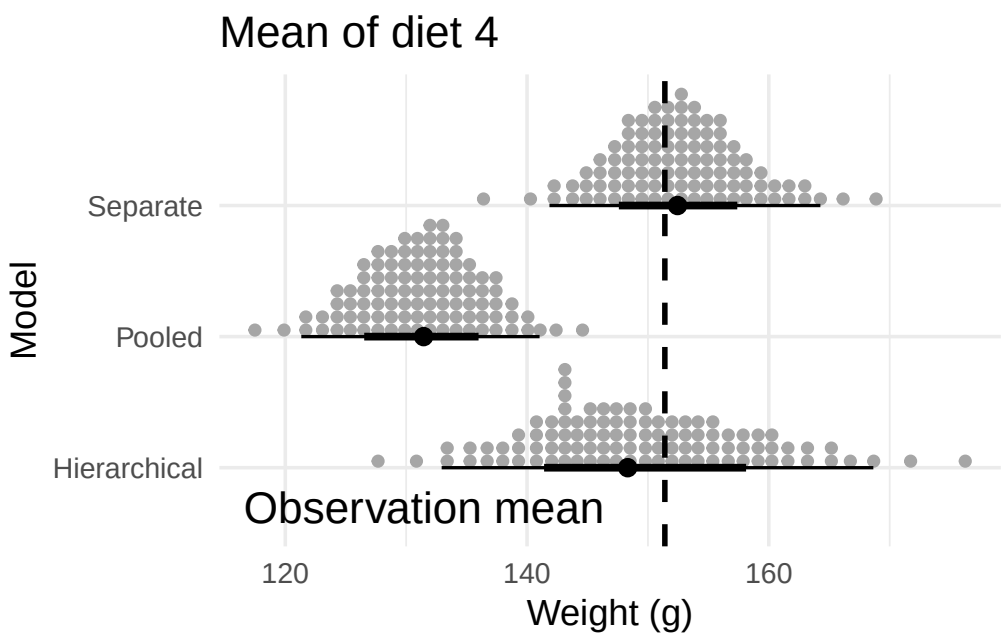


Figure 4: Posterior distribution of the mean weight of chicks consuming diet 4.

Firstly, Both Pooled and Separate follow normal distribution which is expected. Hierarchical has higher variance. In this case, it appears that the Separate model may be slightly overestimating, the Pooled underestimating, and theUse the provided code in the template to plot the posterior distribution of the mean of the weight measurements of the fourth diet and comment on the possible differences you observe between the models.'s estimate is closest to the observed mean, which could suggest that it's providing a more accurate estimate in this particular case. From this small data it seems Hierarchical is the best predictor of deit.

2.20 (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"
  )
```

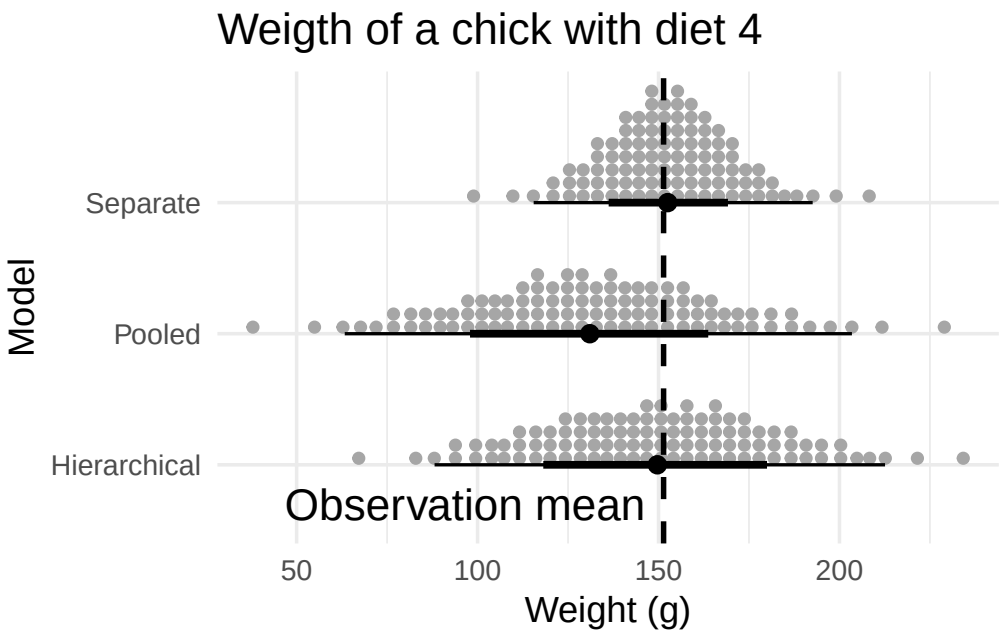


Figure 5: The (posterior) predictive distribution of the weigth of a chick consuming diet 4.

The means for each model are close to th observation mean for all the models, but relatively Pooled seems to be underestimating. The distributions have some differences a with separate with the highest peak and variance. Poole and Hierarchical have similar distributions but as Hierarchical is better postioned around the observation mean, one could conclude Hierarchical to be most sufficient in predicting the weight.

2.21 (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")
```

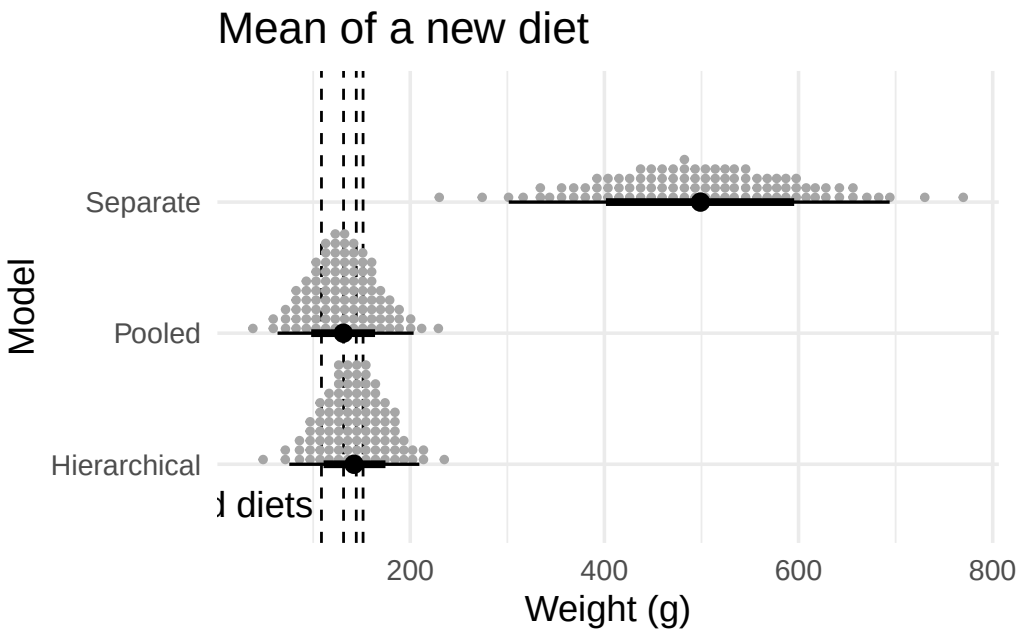


Figure 6: Posterior distribution of the mean weight of chicks consuming the new diet 5 not seen before.

This has the highest difference with Separate is overestimating the weights, in similar way done earlier by informed priors. Therefore an Pooled and Hierarchical seems to be closest to the true distribution. It is hard to determine from this picture which is better Hierarchical or Pooled model.

2.22 (p)

```
quantile_Hierarcical <-quantile2(posterior_mean_diet_4$mean_diet_4[posterior_mean_diet_4$model_name
quantile_Separate <- quantile2(posterior_mean_diet_4$mean_diet_4[posterior_mean_diet_4$model_name ==
quantile_Pooled <- quantile2( posterior_mean_diet_4$mean_diet_4[posterior_mean_diet_4$model_name ==
```

```
quantile_Hierarcical
```

```
      q5       q95
135.3297 165.1727
```

```
quantile_Separate
```

```
      q5       q95
143.7795 162.2433
```

```
quantile_Pooled
```

```
      q5      q95
123.1299 139.4472
```

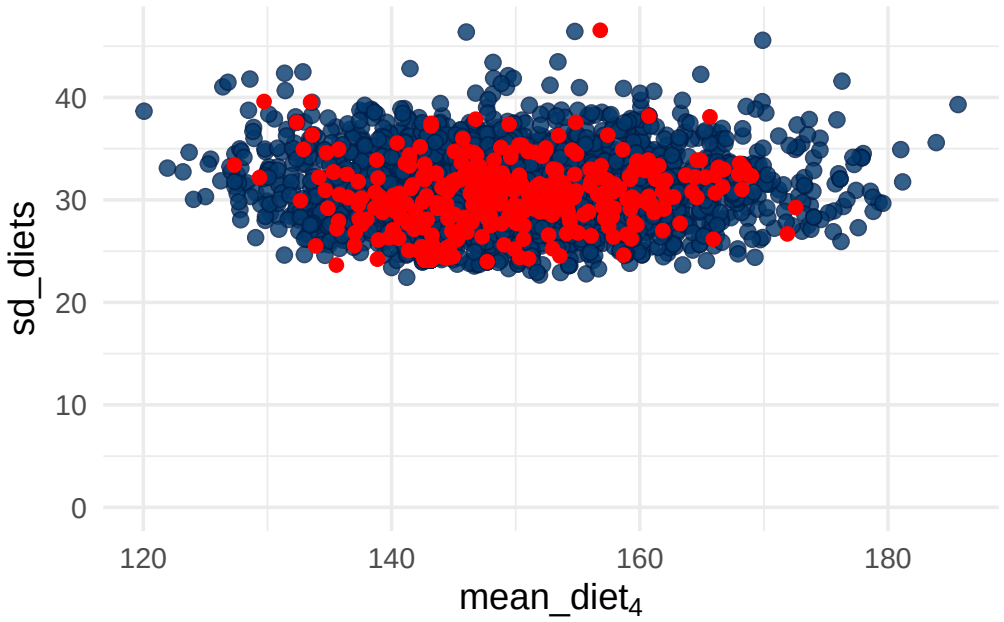
The 90% quantile range for Hierarchical = 134 to 165 The 90% quantile range for Separate = 143 to 162 The 90% quantile range for Pooled = 123 to 139

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.
```



Observing the plot, there is a visible cluster of red dots scattered primarily throughout the main cloud of blue dots. Divergences may be indicated by sudden and large jumps in the parameter values which can be seen with some outliers but in my opinion this seems to have been converged.

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(150,10), class="Intercept"), # prior for mu_0
    # REPLACE WITH YOUR PRIOR FOR SIGMA
    prior(normal(30,5), class="sigma"),     # prior for sigma
    # REPLACE WITH YOUR PRIOR FOR SD
    prior(normal(150,30), class="sd")         # prior for tau
  ),
  backend = "cmdstanr",
  save_pars = save_pars(manual = c("z_1[1,4]"))
)

summary(brms_fit)
```

3.3 (c)

Tau estimate is 172 with an 90% interval of [95, 244] Sigma estimate is 30 with an 90% interval of [26, 36] Mu_0 estimate is 150 with an 90% interval of [130, 168]

The results are very close to the earlier assignment with comparable Mu_0 from the earlier assignment at [134, 165]

```
# Draws for mu_4
mu_4 = posterior_epred(brms_fit, newdata = data.frame(Diet=4))
quantile2( mu_4, probs = c(0.05, 0.95))
```

```
      q5       q95
134.7092 167.5012
```

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

3.4 (d)

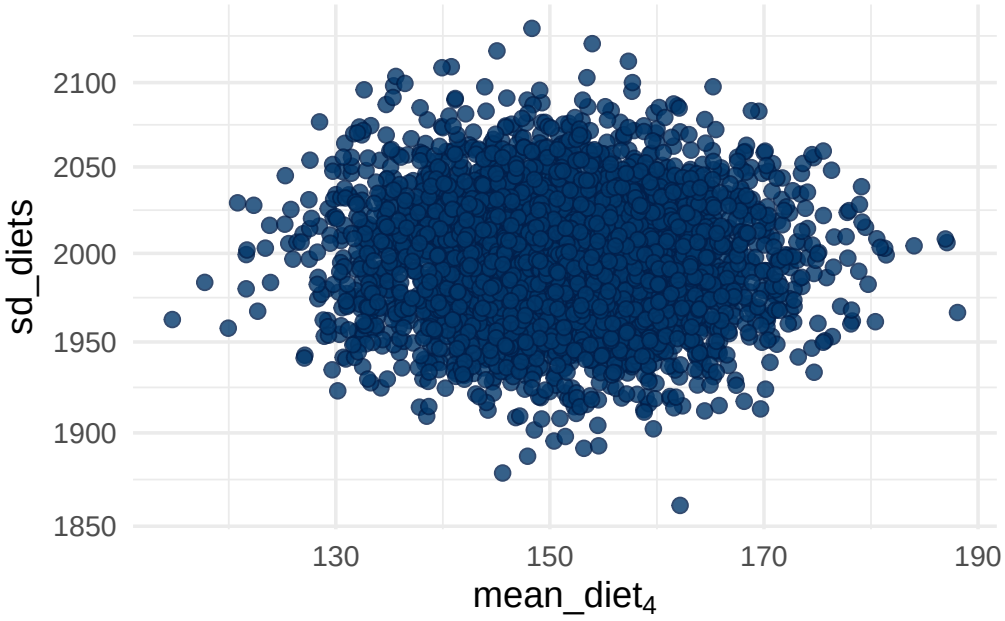
```
#Due the non-centered parametrization, we need to transform compute the $\mu_d$ term as the sum of t



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))
```



The results are worse in this scatter plot than the earlier. There are more outliers which could mean divergence although the summary concluded that the dataset had diverged. Its important to not belive as

3.5 (e)

- The closer one is with the prior the less divergent transitions occurred.
- To get thee result of divergence one has to choose specific extreme values that it will search areas which are irrelevant
- Centered parameterizations can have problems with sampling, especially in hierarchical models where there is a so-called “funnel” problem. When using centered parameterizations, the varying effects can be highly correlated with the hyperparameters, particularly when the group-level standard deviations are small.