

Assignment 8

LOO-CV model comparison

anonymous

1 General information

I did not use AI for solving this exercise.

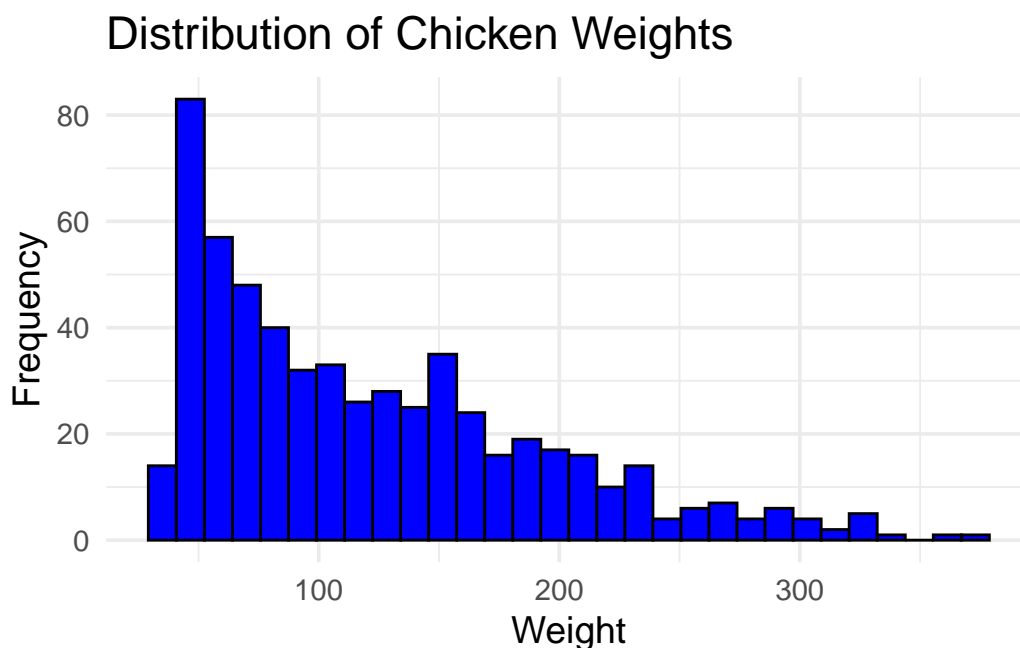
2 A hierarchical model for chicken weight time series

2.1 Exploratory data analysis

2.2 (a)

```
# Useful functions: ggplot, aes(x=...), geom_histogram
data("ChickWeight")
# Create the histogram using ggplot
ggplot(ChickWeight, aes(x = weight)) +
  geom_histogram(fill = "blue", color = "black") +
  ggtitle("Distribution of Chicken Weights") +
  xlab("Weight") +
  ylab("Frequency")
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



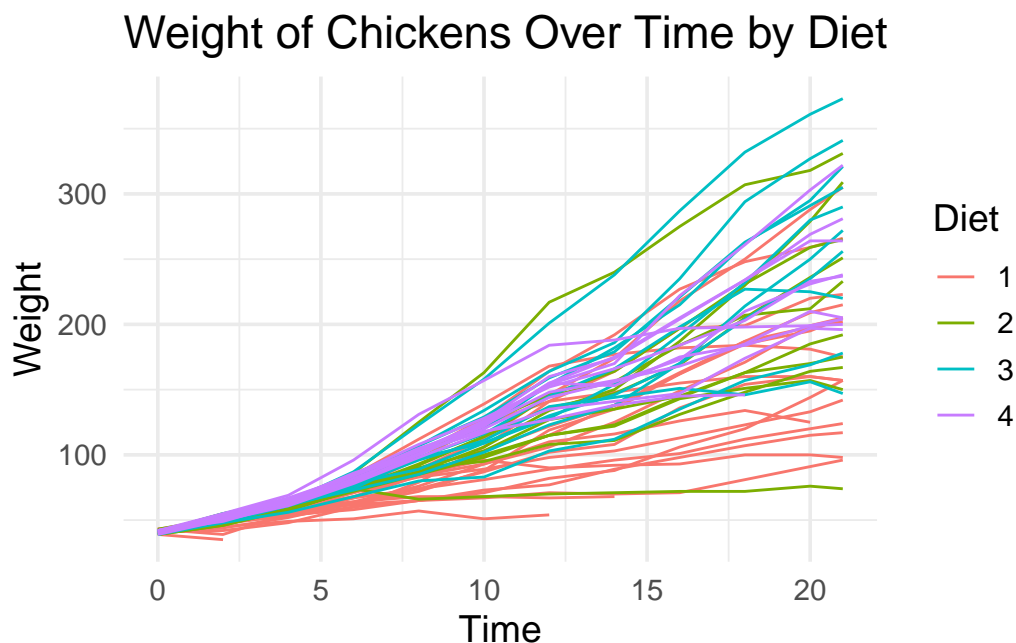
Weight Distribution: The histogram shows that the majority of the chicken weights are clustered in the lower range, suggesting that most chickens weigh less than 100 units. There is a long tail extending towards the heavier weights, indicating that there are relatively fewer chickens at these higher weights.

Qualitative Range: The range of chicken weights seems to be from just above 0 to slightly more than 300 units. The data are skewed to the right, with a peak frequency around the 40-60 units weight range.

The histogram of chicken weights from the 'ChickWeight' dataset shows a right-skewed distribution with most of the data concentrated in the lower weight range, specifically around 40 to 60 units. The frequency of observations decreases as the weight increases, with few chickens weighing over 200 units. This suggests a wide but uneven spread of chicken weights in the dataset.

2.3 (b)

```
# Useful functions: ggplot, aes(x=...,y=...,group=...,color=...), geom_line
ggplot(ChickWeight, aes(x = Time, y = weight, group = Chick, color = factor(Diet))) +
  geom_line() +
  ggtitle("Weight of Chickens Over Time by Diet") +
  xlab("Time") +
  ylab("Weight") +
  scale_color_discrete(name = "Diet")
```



Individual Growth Trajectories: The plot displays individual growth trajectories of chickens over time, differentiated by the diet they were on.

Diet Influence: It is evident that diet plays a role in the growth rates of the chickens. The colors represent different diets, and we can see that some diets result in faster or greater weight gain over time.

Weight Gain Over Time: Generally, there is an increase in weight over time for all diets, which is expected as the chickens grow.

Variation Among Chickens: There's a noticeable variation in weight among chickens even within the same diet group, indicating that factors other than diet may also influence growth.

The line plot illustrates the growth patterns of chickens over time, with each line representing an individual chicken. The use of color to differentiate the diets shows that some diets are associated with more rapid weight gain. While there's an overall trend of weight increase over time across all diets, there's also significant variation within each diet group, suggesting individual differences among chickens or other unaccounted variables influencing growth. The plot is effective in conveying these trends and allows for quick visual comparison of the growth rates associated with different diets.

2.4 Linear regression

2.5 (c)

In `brms`, a regression can be specified as below, see also [below \(#m\)](#) or [the last template](#). Fill in the appropriate variables, data, and likelihood family. Specify the priors, then run the model (by removing `#| eval: false` below).

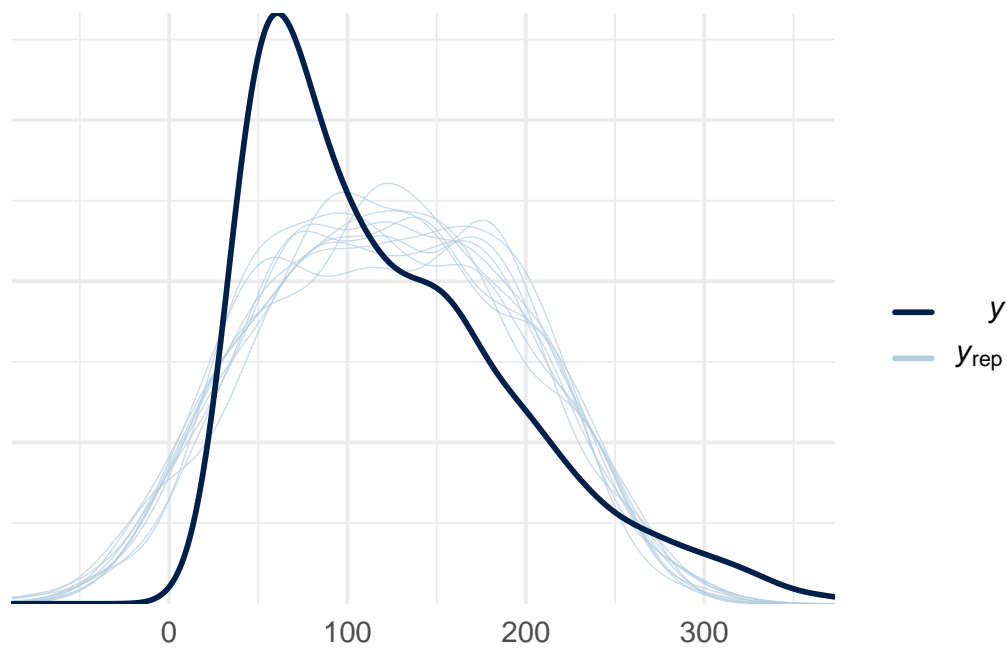
```
priors <- c(
  prior(normal(0, 10), coef = "Time"),
  prior(normal(0, 10), coef = "Diet2"),
  prior(normal(0, 10), coef = "Diet3"),
  prior(normal(0, 10), coef = "Diet4")
)

f1 <- brms::brm(
  # This specifies the formula
  weight ~ 1 + Time + Diet,
  # This specifies the dataset
  data = ChickWeight,
  # This specifies the observation model family
  family = gaussian(),
  # This passes the priors specified above to brms
  prior = priors,
  # This causes brms to cache the results
  #file = "additional_files/assignment8/f1",
  save_pars = save_pars(all = TRUE)
)
```

2.6 (d)

```
# Useful functions: brms::pp_check
pp_check(f1)
```

Using 10 posterior draws for ppc type 'dens_overlay' by default.



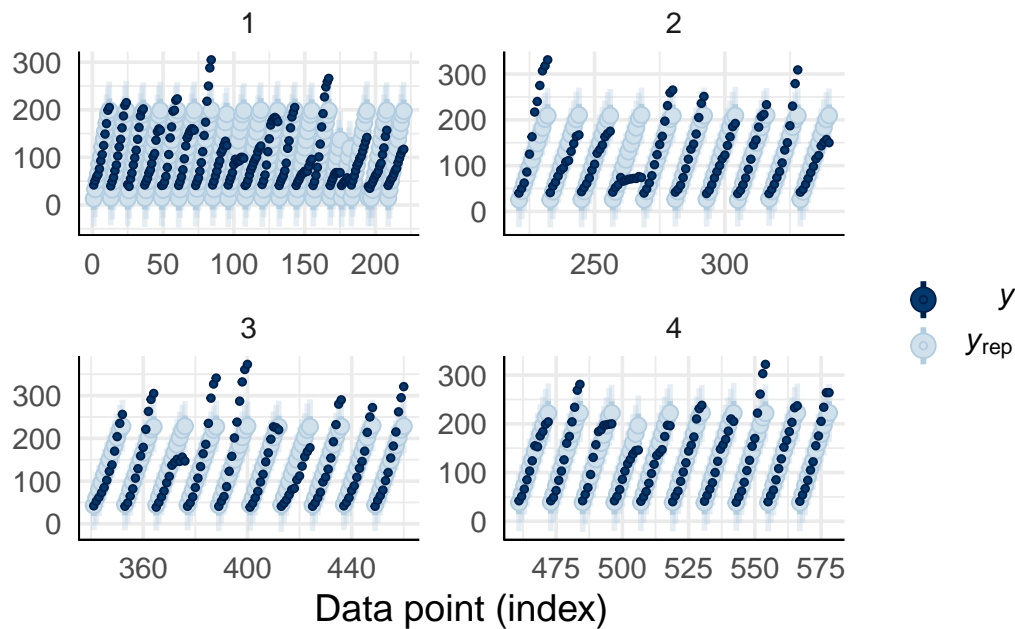
Distribution Shape: The replicated data sets seem to follow the same general shape as the observed data, with a peak and a decline. However, there is **variability** in the predicted data, as expected from posterior predictive checks.

Fit to Observed Data: As we observe the predicted and original datasets, there are differences between them. In the predicted dataset, the given range spans from negative values, whereas the original dataset has a range from zero since weights are positive numbers.

2.7 (e)

```
# Useful functions: brms::pp_check(..., type = ..., group=...)
pp_check(f1, type = "intervals_grouped", group = "Diet")
```

Using all posterior draws for ppc type 'intervals_grouped' by default.



Comparison of Observed and Predicted: The observed data points (y) are plotted alongside the predicted intervals (y_{rep}). In some cases, the observed points fall outside of the predicted intervals, which could indicate areas where the model is not capturing the data well.

Based on the visualizations, if there are consistent patterns where the observed data falls outside the predicted intervals, this suggests room for model improvement. One reasonable way to improve the model could be to consider interactions between ‘Time’ and ‘Diet’, as the effect of time on weight might differ across diets. Additionally, if the variability is not consistent across the range of weights, a model that allows for non-constant variance (such as using a varying-effects model or a model with a non-Gaussian family for the response distribution) could be more appropriate.

When the observed data points systematically fall outside the predicted intervals, it can indicate that the model is missing some aspect of the data structure. For instance, if higher weights are less well-predicted, it could mean that the relationship between diet, time, and weight is not strictly linear or that there are other factors at play that are not accounted for in the model.

To further improve the model, you could also consider:

- Adding polynomial terms for ‘Time’ if the growth rate is not linear.
- Using log-normal family to avoid negativity.
- Including random effects if there is clustering in the data (e.g., chickens within the same batch might be more similar to each other).
- Examining other covariates that might influence weight, such as initial weight, genetic factors, or environmental conditions.

2.8 Log-normal linear regression

2.9 (f)

```
log_priors <- c(
  prior(normal(0, log(3)), coef = "Time"),
  prior(normal(0, log(5)), coef = "Diet2"),
```

```

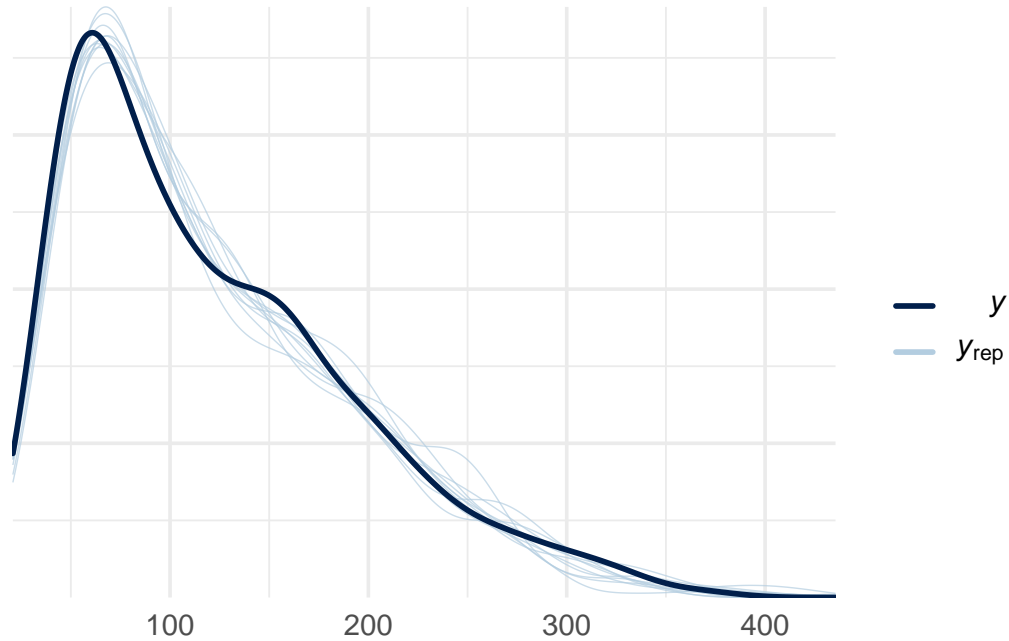
prior(normal(0, log(5)), coef = "Diet3"),
prior(normal(0, log(5)), coef = "Diet4")
)

f2 <- brms::brm(
  weight ~ 1 + Time + Diet,
  data = ChickWeight,
  family = "lognormal",
  prior = priors,
  #file = "additional_files/assignment8/f2",
  save_pars = save_pars(all = TRUE)
)

pp_check(f2)

```

Using 10 posterior draws for ppc type 'dens_overlay' by default.

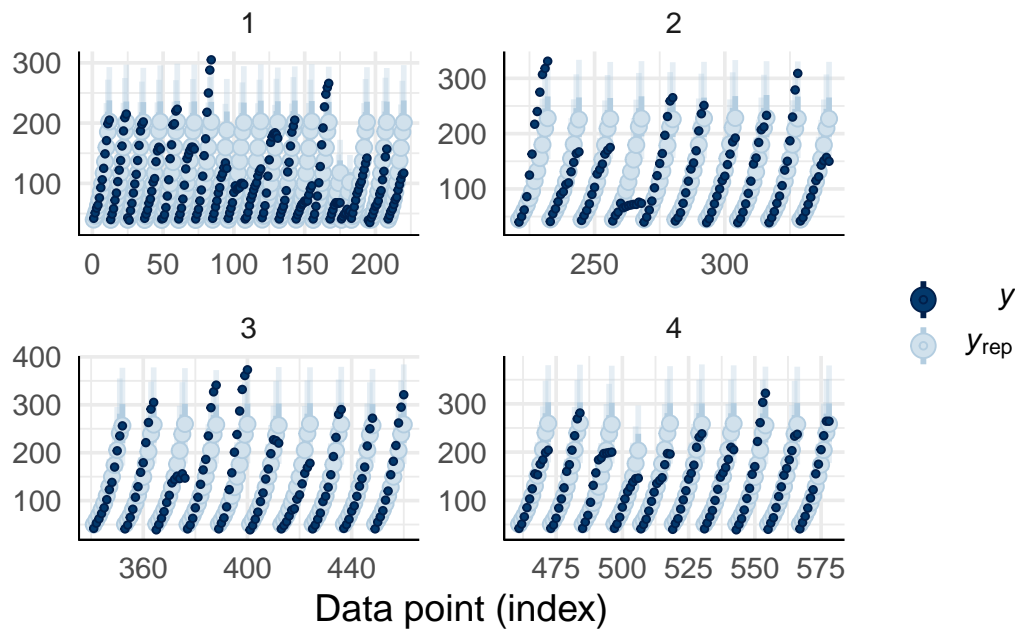


```

pp_check(f2, type = "intervals_grouped", group = "Diet")

```

Using all posterior draws for ppc type 'intervals_grouped' by default.



The switch to a lognormal distribution seems to have improved the model fit, especially for the higher weights where a normal model might have underestimated the variance.

The lognormal model's multiplicative nature is likely more appropriate for growth data, which tend to be skewed and not symmetrically distributed.

Some data points still fall outside the predicted intervals, indicating potential areas for further model improvement, such as considering interactions between predictors, nonlinear effects, or heteroscedasticity.

Overall, the updated model with a lognormal distribution appears to better capture the distribution of the observed data, particularly in terms of the spread and skewness, which are common characteristics of biological growth data.

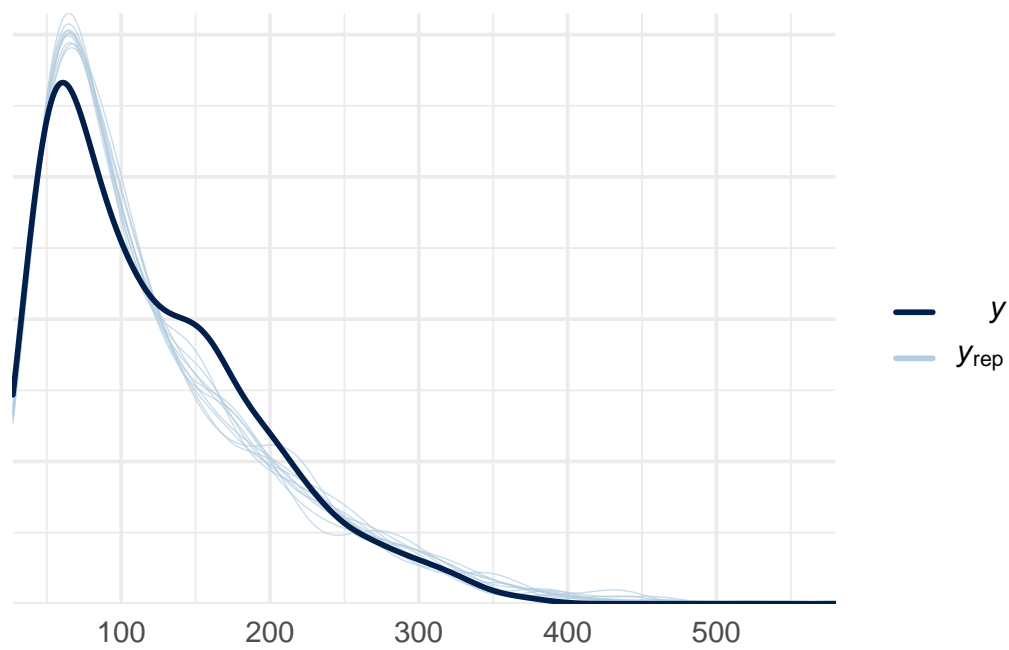
2.10 Hierarchical log-normal linear regression

2.11 (g)

```
f3 <- brms::brm(
  weight ~ 1 + Time + Diet + (Time | Chick),
  data = ChickWeight,
  family = "lognormal",
  prior = priors,
  #file = "additional_files/assignment8/f3",
  save_pars = save_pars(all = TRUE)
)
```

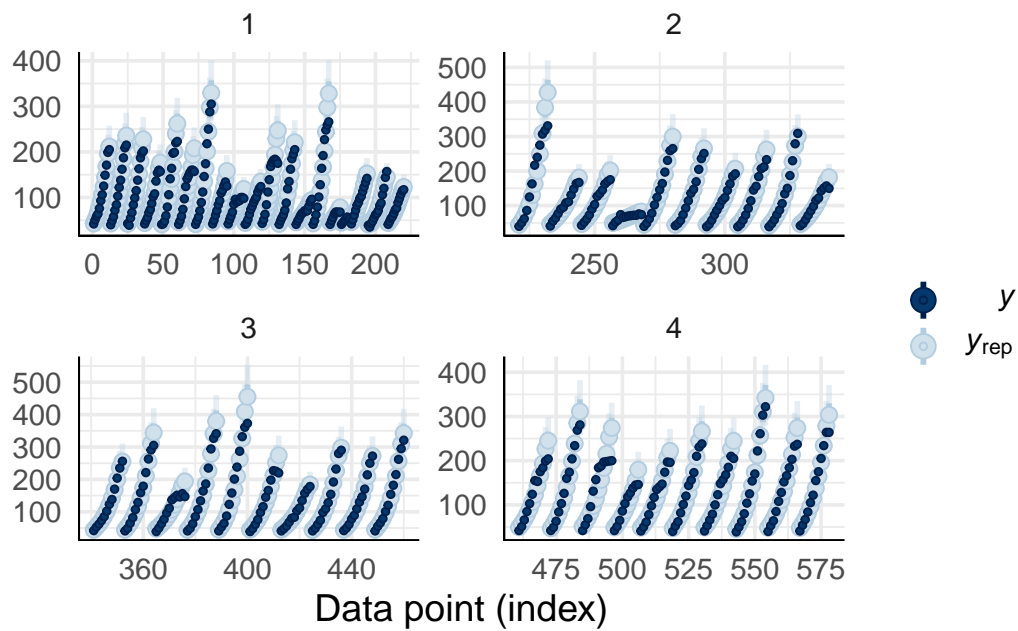
```
pp_check(f3)
```

Using 10 posterior draws for ppc type 'dens_overlay' by default.



```
pp_check(f3, type = "intervals_grouped", group = "Diet")
```

Using all posterior draws for ppc type 'intervals_grouped' by default.



Looking at the charts, it's clear that the predicted data matches the original data quite closely, showing a more accurate trend. The data is spread out moderately and doesn't show any obvious overfitting or negative patterns.

2.12 (h)

2.13 Model comparison using the ELPD

```
summary(f1)
```

```
Family: gaussian
Links: mu = identity; sigma = identity
Formula: weight ~ 1 + Time + Diet
Data: ChickWeight (Number of observations: 578)
Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
       total post-warmup draws = 4000
```

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	14.68	3.27	8.27	21.02	1.00	4433	2924
Time	8.76	0.23	8.30	9.19	1.00	4504	2959
Diet2	11.16	3.76	3.64	18.38	1.00	4641	3075
Diet3	29.44	3.86	21.94	37.01	1.00	4062	2962
Diet4	23.75	3.79	16.40	31.24	1.00	4320	2831

Family Specific Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	36.19	1.09	34.18	38.44	1.00	4908	2999

Draws were sampled using `sample(hmc)`. 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).

```
summary(f2)
```

```
Family: lognormal
Links: mu = identity; sigma = identity
Formula: weight ~ 1 + Time + Diet
Data: ChickWeight (Number of observations: 578)
Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
       total post-warmup draws = 4000
```

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	3.68	0.02	3.64	3.72	1.00	3861	3579
Time	0.08	0.00	0.07	0.08	1.00	5028	3080
Diet2	0.12	0.03	0.07	0.17	1.00	3194	2979
Diet3	0.25	0.03	0.20	0.30	1.00	2910	2743
Diet4	0.25	0.03	0.20	0.30	1.00	2982	3002

Family Specific Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	0.23	0.01	0.22	0.24	1.00	3395	2201

Draws were sampled using `sample(hmc)`. 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).

```
summary(f3)
```

```
Family: lognormal
Links: mu = identity; sigma = identity
Formula: weight ~ 1 + Time + Diet + (Time | Chick)
Data: ChickWeight (Number of observations: 578)
Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
       total post-warmup draws = 4000
```

Group-Level Effects:

~Chick (Number of levels: 50)

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sd(Intercept)	0.06	0.01	0.03	0.09	1.00	1237	1177
sd(Time)	0.02	0.00	0.02	0.03	1.00	1199	2234
cor(Intercept,Time)	-0.60	0.17	-0.87	-0.20	1.00	514	677

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	3.76	0.02	3.73	3.80	1.00	2183	2480
Time	0.08	0.00	0.07	0.08	1.00	1489	2182
Diet2	0.05	0.03	0.00	0.11	1.00	2254	2804
Diet3	0.06	0.03	-0.00	0.12	1.00	1778	2302
Diet4	0.13	0.03	0.07	0.18	1.00	1948	2660

Family Specific Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	0.11	0.00	0.10	0.11	1.00	3208	3105

Draws were sampled using `sample(hmc)`. 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).

In summary, the models appear to have converged well based on the Rhat values (all of them are less than 1.1), and the effective sample sizes are generally sufficient, although some parameters, especially in the hierarchical model, might need further investigation. There's no explicit mention of divergent transitions, which is typically a positive sign. This overall suggests that the models are reliable from a convergence standpoint.

2.14 (i)

```
# Useful functions: loo, loo_compare
mmlloo1 <- loo(f1, moment_match = TRUE)
#mmlloo1 <- loo_moment_match(f1, loo = loo1)
mmlloo2 <- loo(f2, moment_match = TRUE)
#mmlloo2 <- loo_moment_match(f2, loo = loo2)
mmlloo3 <- loo(f3, moment_match = TRUE)
#mmlloo3 <- loo_moment_match(f3, loo = loo3)
```

```
# Compare models
loo_comparison <- loo_compare(mmlloo1, mmlloo2, mmlloo3)
print(loo_comparison)
```

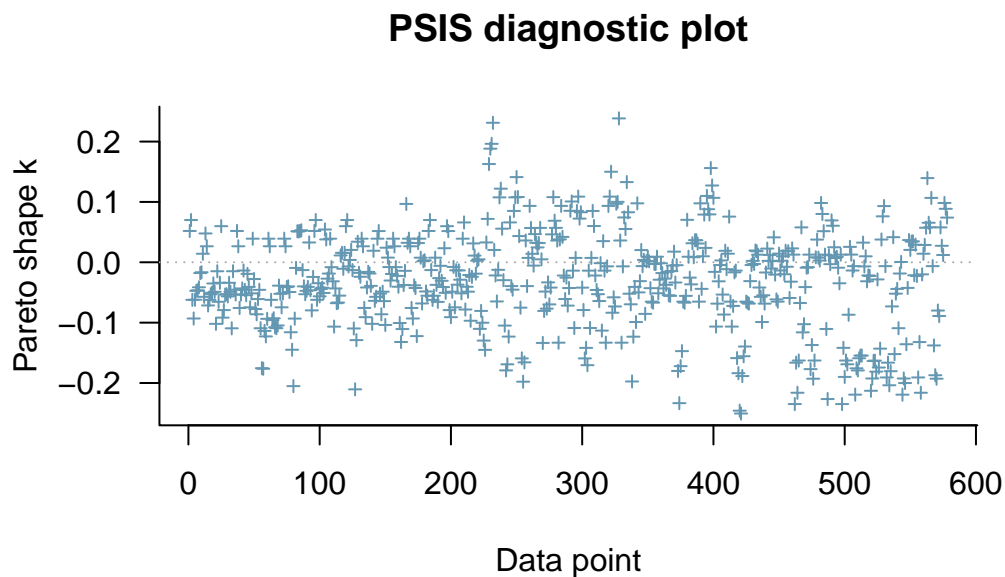
```
      elpd_diff se_diff
f3      0.0      0.0
f2 -394.6     30.2
f1 -642.8     32.1
```

Best Predictive Performance: The model with the best predictive performance is indicated by the highest expected log predictive density (ELPD). In your case, f3 (the hierarchical log-normal linear regression model) has the highest ELPD, as it has an `elpd_diff` of 0.0 compared to the other models. This indicates that f3 is the best model in terms of predictive performance among the three models compared.

Influence of Uncertainty: These substantial differences in ELPD, accompanied by relatively smaller standard errors, suggest that the uncertainty does not significantly influence the decision of which model is best. f3 remains the best model with considerable confidence.

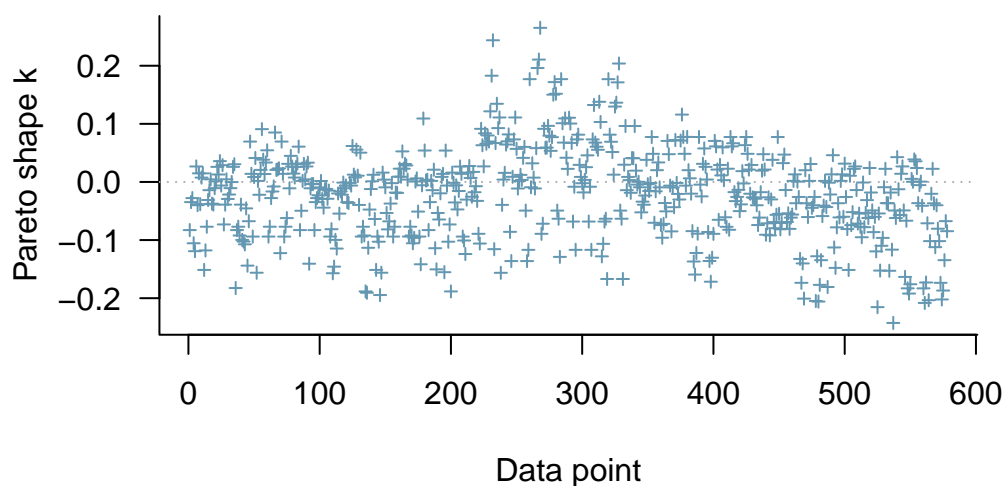
2.15 (j)

```
# Useful functions: plot(loo(...), label_points = TRUE)
plot(mmlloo1, label_points = TRUE)
```



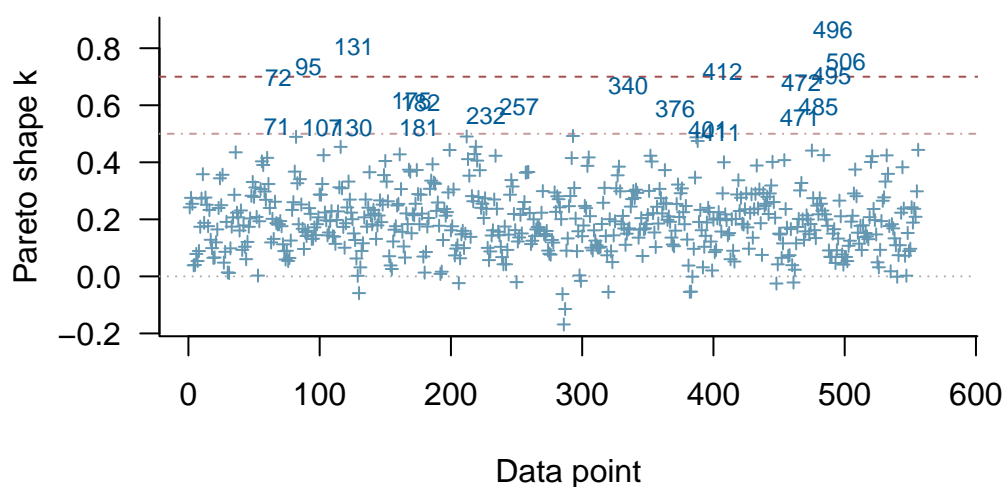
```
plot(mmlloo2, label_points = TRUE)
```

PSIS diagnostic plot



```
plot(mmlloo3, label_points = TRUE)
```

PSIS diagnostic plot



Observing the plots, it's evident that the hierarchical log-normal model has a higher estimated value of \hat{k} compared to the others.

2.16 (k)

```
# Assuming you have a hierarchical model fit
# Select chickens with high and low Pareto k values
k_hat_values <- mmlloo3$diagnostics$pareto_k
high_k_chickens <- ChickWeight$Chick[order(k_hat_values, decreasing = TRUE)[1:4]]
```

```
low_k_chickens <- ChickWeight$Chick[order(k_hat_values)[1:4]] # Chickens with low k values
print("Chicken with high k hat value:")
```

```
[1] "Chicken with high k hat value:"
```

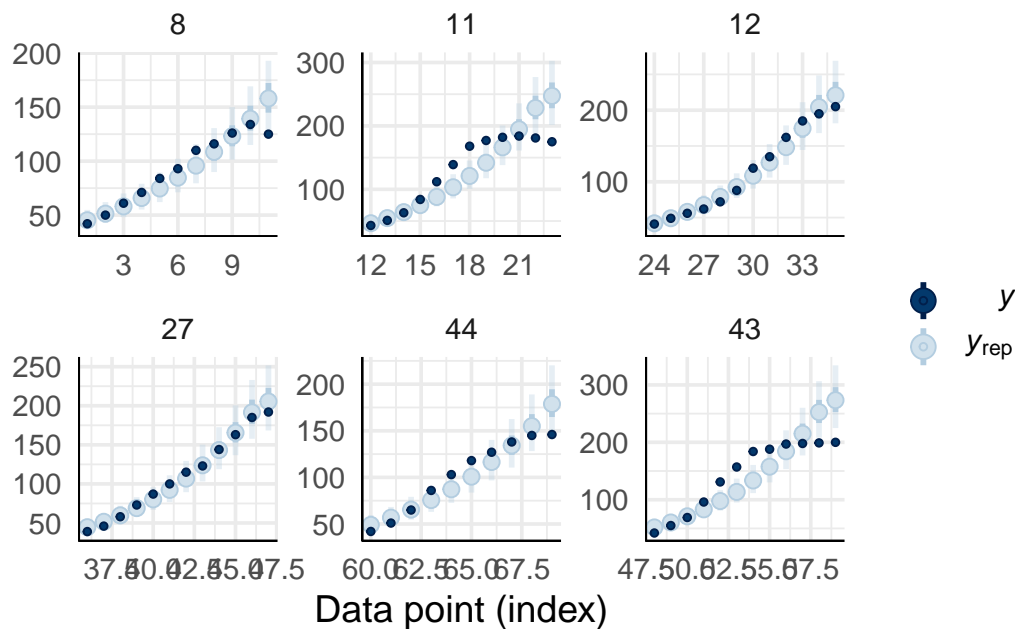
```
print(high_k_chickens)
```

```
[1] 43 11 44 8
```

```
50 Levels: 18 < 16 < 15 < 13 < 9 < 20 < 10 < 8 < 17 < 19 < 4 < 6 < 11 < ... < 48
```

```
# PPC for selected chickens
brms::pp_check(
  f3,
  type = "intervals_grouped",
  group = "Chick",
  newdata = ChickWeight |> filter(Chick %in% c(high_k_chickens, low_k_chickens))
)
```

Using all posterior draws for ppc type 'intervals_grouped' by default.



The graphs suggest that the model performs more effectively with lower \hat{k} values than with higher ones, showcasing a superior fit in the former case. The model demonstrates enhanced performance when \hat{k} is on the lower side.

2.17 Model comparison using the RMSE

2.18 (I)

```
rmse <- function(fit, use_loo = FALSE) {
  mean_y_pred <- if(use_loo) {
    brms::loo_predict(fit)
  } else {
    colMeans(brms::posterior_predict(fit))
  }
  sqrt(mean((mean_y_pred - brms::get_y(fit))^2))
}

# Compute RMSE and LOO-RMSE for your models
rmse_fit1 <- rmse(f1)
loo_rmse_fit1 <- rmse(f1, use_loo = TRUE)
```

Running PSIS to compute weights

Warning: 'log_ratios' not specified. Can't compute k-hat diagnostic.

```
print(paste("RMSE for model fit1: ", rmse_fit1))
```

```
[1] "RMSE for model fit1: 35.9770944161715"
```

```
print(paste("LOO-RMSE for model fit1: ", loo_rmse_fit1))
```

```
[1] "LOO-RMSE for model fit1: 36.2612178041343"
```

```
rmse_fit2 <- rmse(f2)
loo_rmse_fit2 <- rmse(f2, use_loo = TRUE)
```

Running PSIS to compute weights

Warning: 'log_ratios' not specified. Can't compute k-hat diagnostic.

```
print(paste("RMSE for model fit2: ", rmse_fit2))
```

```
[1] "RMSE for model fit2: 34.742370128816"
```

```
print(paste("LOO-RMSE for model fit2: ", loo_rmse_fit2))
```

```
[1] "LOO-RMSE for model fit2: 35.1339241581717"
```

```
rmse_fit3 <- rmse(f3)
loo_rmse_fit3 <- rmse(f3, use_loo = TRUE)
```

Running PSIS to compute weights

Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.

Warning: 'log_ratios' not specified. Can't compute k-hat diagnostic.

```
print(paste("RMSE for model fit3: ", rmse_fit3))
```

```
[1] "RMSE for model fit3: 15.8350526026849"
```

```
print(paste("LOO-RMSE for model fit3: ", loo_rmse_fit3))
```

```
[1] "LOO-RMSE for model fit3: 19.9468811701026"
```

Difference Between RMSE and LOO-RMSE: - RMSE (Root Mean Square Error) measures the average magnitude of the errors between the predictions and the actual observed values, without considering model complexity or overfitting. It's a straightforward measure of prediction accuracy on the training data.

- LOO-RMSE (Leave-One-Out RMSE) incorporates cross-validation (specifically, leave-one-out cross-validation). It assesses how well the model predicts new, unseen data. By leaving out one observation at a time from the dataset, it estimates how well the model performs on data it was not trained on, thus providing a more robust measure of model predictive performance.

Comparison of RMSE and LOO-RMSE Values: - For all three models (f1, f2, f3), the LOO-RMSE is higher than the RMSE. This is generally expected because RMSE is calculated on the same data that the model was trained on, leading to potentially lower error rates due to model fitting to this specific dataset. In contrast, LOO-RMSE provides a more conservative and arguably more realistic estimate of the model's predictive accuracy on unseen data, as it simulates the model's performance on new data by excluding each observation one at a time during validation.

Model Comparison: - Among the three models, the Hierarchical log-normal linear regression (f3) has the lowest RMSE and LOO-RMSE, indicating its superior predictive accuracy both on the training data and in a cross-validated setting.

- The Log-normal linear regression (f2) and the Linear regression (f1) models show higher RMSE and LOO-RMSE values compared to f3, with f1 performing the least effectively.