

# Assignment 8

## LOO-CV model comparison

anonymous

### General information

AI was used in establishing the f2 and f3 models.

### A hierarchical model for chicken weight time series

#### Exploratory data analysis

(a)

```
head(ChickWeight, 10)
```

```
## Grouped Data: weight ~ Time | Chick
```

```
##   weight Time Chick Diet
```

```
## 1     42    0     1    1
```

```
## 2     51    2     1    1
```

```
## 3     59    4     1    1
```

```
## 4     64    6     1    1
```

```
## 5     76    8     1    1
```

```
## 6     93   10     1    1
```

```
## 7    106   12     1    1
```

```
## 8    125   14     1    1
```

```
## 9    149   16     1    1
```

```
## 10    171  18     1    1
```

```
# Useful functions: ggplot, aes(x=...), geom_histogram
```

```
ggplot(ChickWeight, aes(x = weight)) +
```

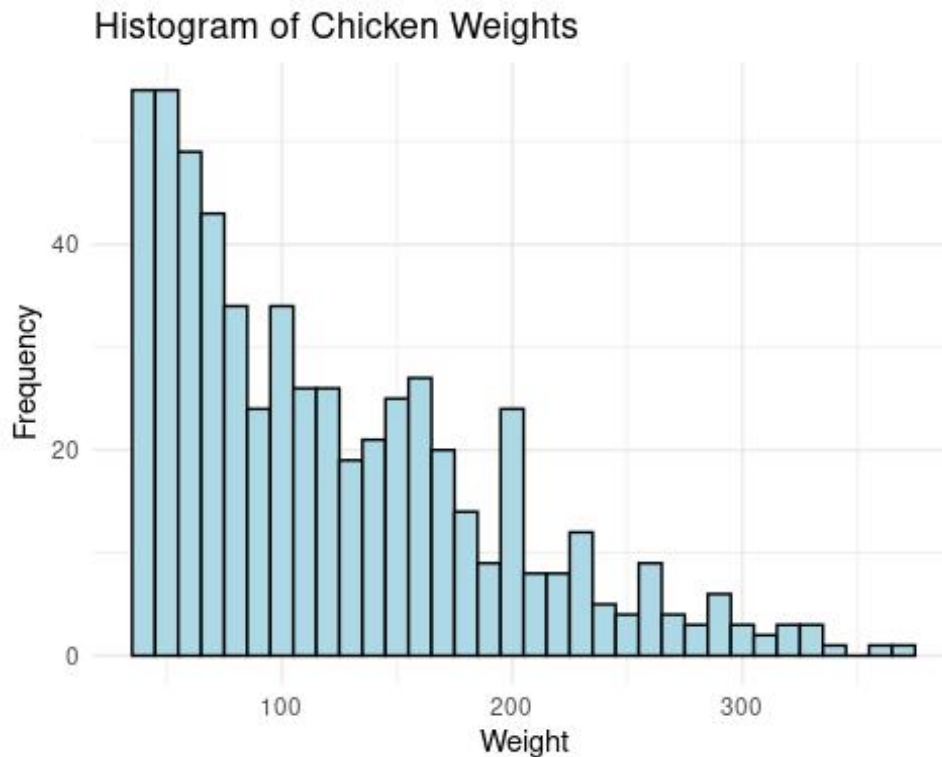
```
  geom_histogram(binwidth = 10, fill = "lightblue", color = "black") +
```

```
  labs(title = "Histogram of Chicken Weights",
```

```
        x = "Weight",
```

```
        y = "Frequency") +
```

```
  theme_minimal()
```



From the plot, we can see that the weights are distributed across a range from the 30 to the 370. There are more observations that are lower in weight and fewer observations that are higher in weight. The weight classes increase as the histogram moves to the right, indicating that chickens tend to be lighter in this particular subset of the data, with fewer chickens reaching the higher weight ranges.

(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() +
  labs(title = "Chicken Weights Over Time by Diet",
       x = "Time (days)",
       y = "Weight (grams)",
       color = "Diet") +
  theme_minimal() +
  scale_color_brewer(palette = "Set1")
```



The line plot visualizes the weight progression of chickens over a period of approximately 20 days, with color coding to represent four different diets. It illustrates that all chickens gain weight over time, but the rate of growth varies both within and between diet groups. Some diets appear to promote faster growth, as indicated by steeper lines, particularly for Diet 3 (green) and possibly Diet 1 (red). There is noticeable variability among individual chickens on the same diet, suggesting that factors beyond diet, such as genetics or health, may also play a role in growth rates. Overlaps and crossings of lines between different diet groups indicate that no single diet is universally effective for all chickens. The plot also shows that while most chickens exhibit a steady increase in weight, some have less pronounced growth, highlighting the presence of outliers or exceptions in the dataset.

## Linear regression

(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, 20), coef = "Time"),
  prior(normal(0, 20), coef = "factorDiet2"),
  prior(normal(0, 20), coef = "factorDiet3"),
```

```

prior(normal(0, 20), coef = "factorDiet4")
)

f1 <- brms::brm(
  # This specifies the formula
  weight ~ 1 + Time + factor(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 = "~/notebooks/Bayes EX8/f1"
)

```

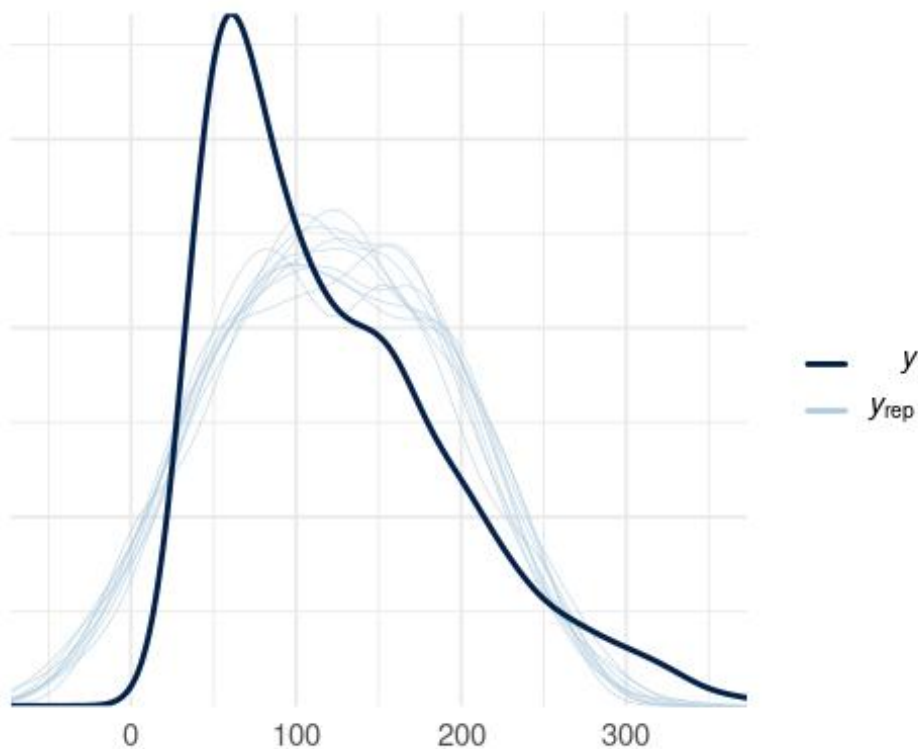
(d)

```

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

```

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



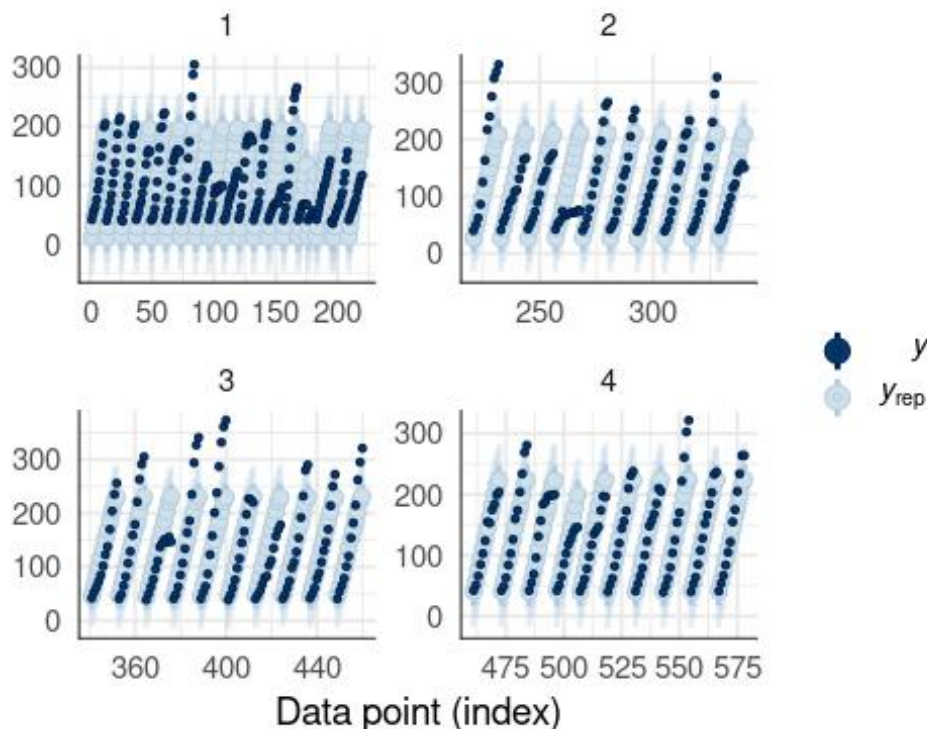
The density of the observed data and the density of the posterior prediction overlap well in some areas, especially in the central part of the data. This means that the model is able to capture major trends in the data. But at the tail of the data, especially the right tail, there seems to be some discrepancy between the

observational data and the posterior prediction. This indicates that the model may not fully capture the tail of the distribution of the data. The light lines show the changes in predictions under different posterior samples, which reflect the uncertainty of the model for future observations. If the lines are very scattered, there is a high degree of uncertainty in the model predictions.

(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.
```



While most of the observed data points fall within the forecast interval, there are still some points that fall outside the interval. These points may represent extreme values or outliers that the model may fail to capture. The distribution of predicted intervals may vary by Diet, which may indicate that different diets have different effects on the weight of chickens, or that the variability of the data varies by diet group.

The method to improve the model: It supposed tp include more predictors, If certain variables that could potentially affect the outcome (such as different diet compositions, environment, age of chickens, etc.) are not currently in the model, including them could improve the fit.

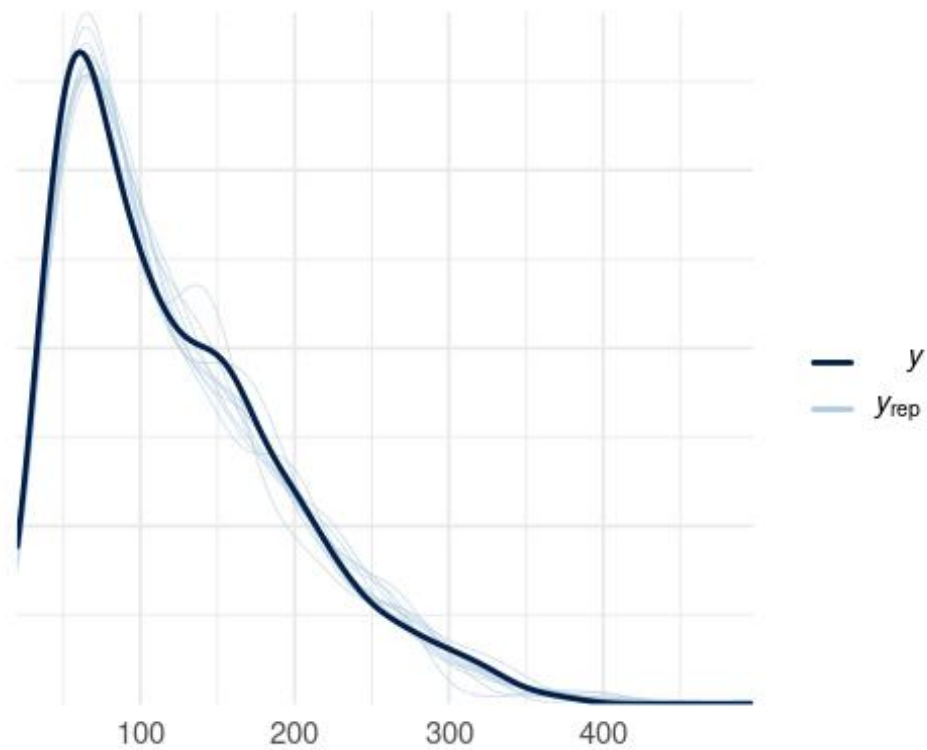
## Log-normal linear regression

(f)

```
log_priors <- c(
  prior(normal(0, log(3)), coef = "Time"),
  prior(normal(0, log(5)), coef = "factorDiet2"),
  prior(normal(0, log(5)), coef = "factorDiet3"),
  prior(normal(0, log(5)), coef = "factorDiet4")
)
# Check the priors that correspond to the model parameters
#prior_info <- get_prior(weight ~ 1 + Time + factor(Diet), data = ChickW
eight, family = gaussian())
#print(prior_info)

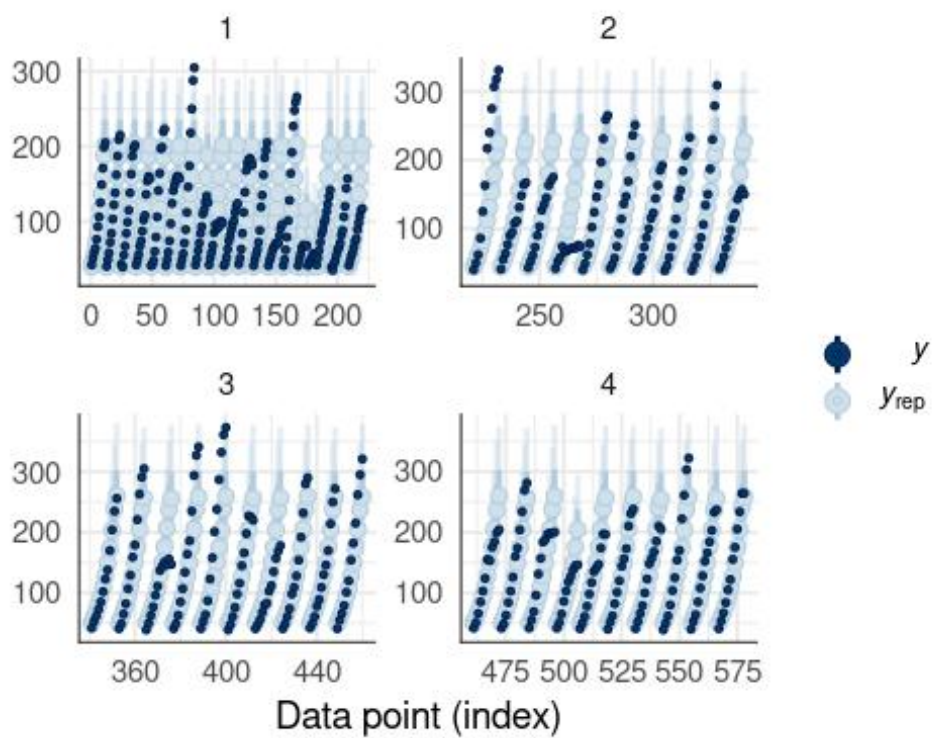
f2 <- brms::brm(
  # This specifies the formula
  formula = weight ~ 1 + Time + factor(Diet),
  # This specifies the dataset
  data = ChickWeight,
  # This specifies the observation model family
  family = lognormal(),
  # This passes the priors specified above to brms
  prior = log_priors,
  # This causes brms to cache the results
  file = "~/notebooks/Bayes EX8/f2"
)
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.
```

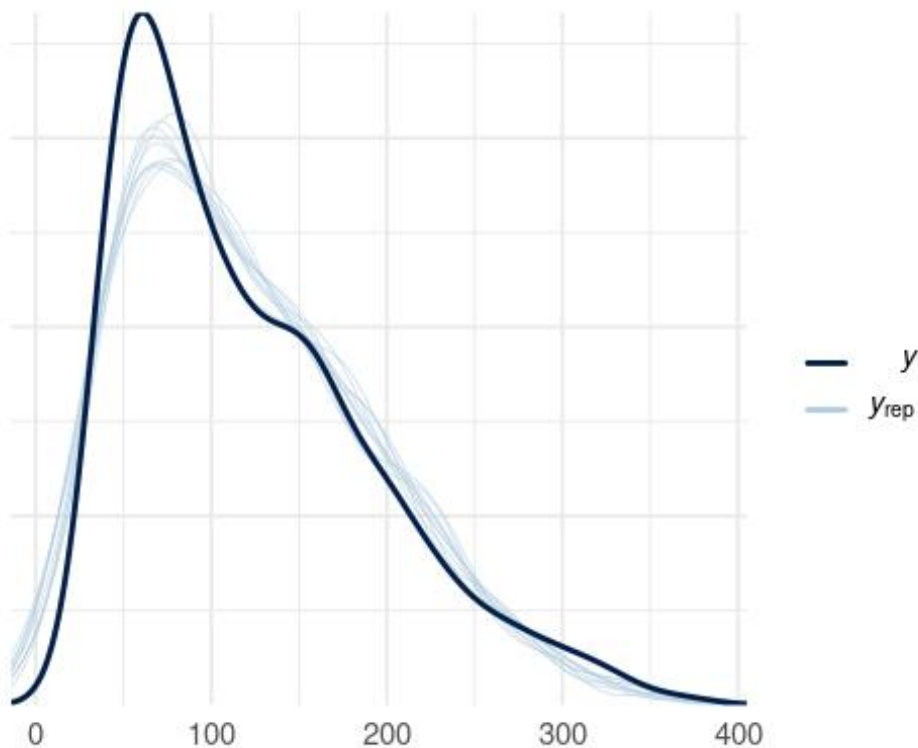


The plot shows a good overlap between the observed data and the simulated posterior predictions, indicating that the model captures the central tendency of the data well, which means the new model better capture some aspects of the data.

## Hierarchical log-normal linear regression

(g)

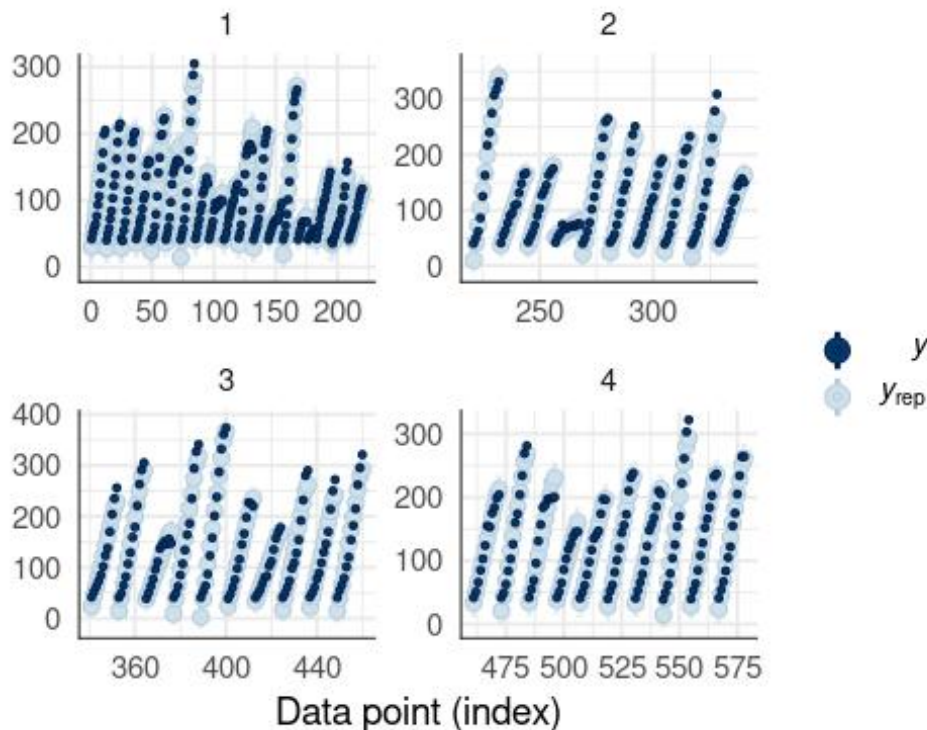
```
priors_g <- c(  
  prior(normal(0, log(3)), coef = "Time"),  
  prior(normal(0, log(5)), coef = "Diet2"),  
  prior(normal(0, log(5)), coef = "Diet3"),  
  prior(normal(0, log(5)), coef = "Diet4")  
)  
#get_prior(weight ~ 1 + Time + Diet, data = ChickWeight, family = gaussian())  
  
f3 <- brms::brm(  
  formula = weight ~ 1 + Time + Diet+(Time | Chick),  
  data = ChickWeight,  
  family = gaussian(),  
  prior = priors_g,  
  file = "~/notebooks/Bayes EX8/f3"  
)  
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.
```



From the first graph we can see that there is a better overlap between the observational data and the model's posteriori prediction data compared with graph in f1, indicating that the model as a whole is able to capture the distribution of the data well. However, the second grouping interval plot shows some differences between the data points grouped by diet and the predicted interval. Although the prediction interval contains most of the observed data points, there are still some points that fall outside the interval, suggesting that the model may not fully capture variability within individual groups. But the model is still good one to fit.

(h)

```
summary(f1)
```

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: weight ~ 1 + Time + factor(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      12.02      3.39      5.49      18.65 1.00      5015      3044
## Time           8.75      0.22      8.33      9.17 1.00      5119      2940
## factorDiet2    14.63      4.01      6.75     22.37 1.00      4725      3243
## factorDiet3    34.34      4.05     26.53     42.43 1.00      4781      3231
## factorDiet4    28.28      4.05     20.49     36.16 1.00      4713      3095
```

```
##
## Family Specific Parameters:
##      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma      36.07      1.07     34.00     38.23 1.00      5636      2963
##
## Draws were sampled using sample(hmc). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the pote
ntial
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

**summary(f2)**

```
## Family: lognormal
## Links: mu = identity; sigma = identity
## Formula: weight ~ 1 + Time + factor(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      3211      3259
## Time           0.08      0.00      0.07      0.08 1.00      4998      3107
## factorDiet2     0.12      0.03      0.07      0.17 1.00      2563      2748
## factorDiet3     0.25      0.03      0.20      0.30 1.00      2927      2997
## factorDiet4     0.25      0.03      0.20      0.30 1.00      2872      2918
```

```
##
## 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      3778      2783
##
## Draws were sampled using sample(hmc). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the pote
ntial
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

**summary(f3)**

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: weight ~ 1 + Time + Diet + (1 + 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
Tail_ESS
## sd(Intercept)      13.71      2.17    10.01    18.30 1.00      1200
      2077
## sd(Time)           4.38      0.57     3.42     5.62 1.00      1012
      1713
## cor(Intercept,Time) -0.96      0.03    -1.00    -0.89 1.01       983
      1349
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept      35.30      2.47    30.75    40.44 1.00      1213      2007
## Time           6.37      0.68     4.90     7.59 1.00      1008      1495
## Diet2          0.12      1.36    -2.57     2.83 1.00      6831      2831
## Diet3         -0.43      1.40    -3.16     2.36 1.00      6588      2910
## Diet4          2.36      1.40    -0.39     5.08 1.00      4783      2659
##
## Family Specific Parameters:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma      12.81      0.42    12.03    13.70 1.00      4656      2353
##
## Draws were sampled using sample(hmc). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the pote
ntial
## scale reduction factor on split chains (at convergence, Rhat = 1).

```

The first model (Gaussian family, no random effects): the Rhat value of all parameters is 1.00, and both Bulk\_ESS and Tail\_ESS are above several thousand, indicating that the model converges well.

The second model (lognormal family, no random effects): It also shows good convergence, with Rhat values of 1.00 for all parameters and an effective sample size of several thousand or more.

The third model (Gaussian family, including random effects) : Although the Rhat value is 1.00, the lower Bulk\_ESS and Tail\_ESS, especially for the random effects part (such as the Bulk\_ESS of cor(Intercept,Time) being only 668), may mean that estimates of these parameters are not as robust as those for fixed effects. This may be due to the increased complexity of the model, or the need for more iterations to ensure adequate exploration of the parameter space.

In general, the parameters of the model seem to converge quite well, with no obvious convergence problems. However, for models that include random effects, more attention may be needed to ensure that the estimates for the random effects part are also reliable.

## Model comparison using the ELPD

(i)

*# Useful functions: loo, loo\_compare*

```
library(loo)
```

```
## This is loo version 2.6.0
```

```
## - Online documentation and vignettes at mc-stan.org/loo
```

```
## - As of v2.0.0 loo defaults to 1 core but we recommend using as many  
as possible. Use the 'cores' argument or set options(mc.cores = NUM_CORES)  
for an entire session.
```

```
loo1 <- loo(f1)
```

```
loo2 <- loo(f2)
```

```
loo3 <- loo(f3)
```

```
## Warning: Found 4 observations with a pareto_k > 0.7 in model 'f3'. It  
is
```

```
## recommended to set 'moment_match = TRUE' in order to perform moment  
matching
```

```
## for problematic observations.
```

```
loo_compare(loo3,loo2,loo1)
```

```
##      elpd_diff se_diff
```

```
## f3      0.0      0.0
```

```
## f2 -312.3     34.8
```

```
## f1 -558.7     30.5
```

Model f3 performs best in predictive performance because its elpd\_diff value is set to 0.0 for the benchmark. The predictive performance of models f2 and f1 is worse than that of f3.

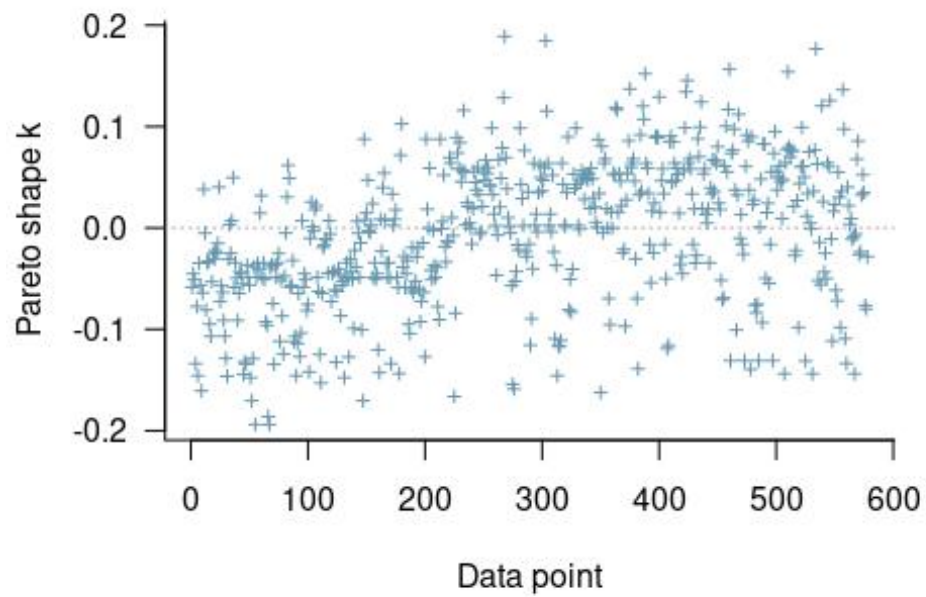
In the result, the elpd\_diff between model f3 and f2 is -312.3, while the se\_diff is 34.8. This means that while there is a significant performance difference, because se\_diff is relatively small, the difference is credible. Similarly, the elpd\_diff between model f3 and f1 is -558.7 and the se\_diff is 30.5, which is also a statistically significant difference because the difference is much larger than the standard error.

(j)

*# Useful functions: plot(loo(...), label\_points = TRUE)*

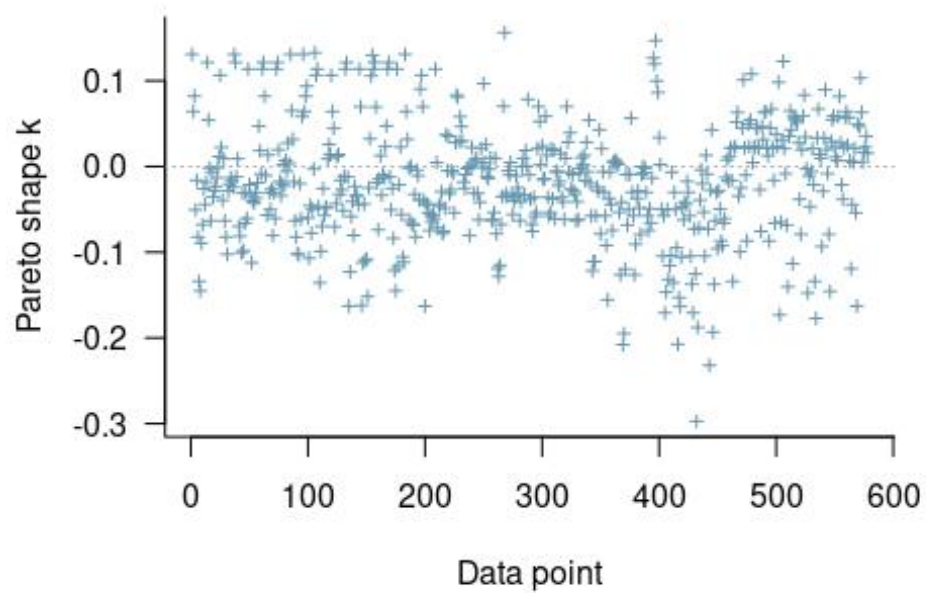
```
plot(loo1,label_points = TRUE)
```

**PSIS diagnostic plot**

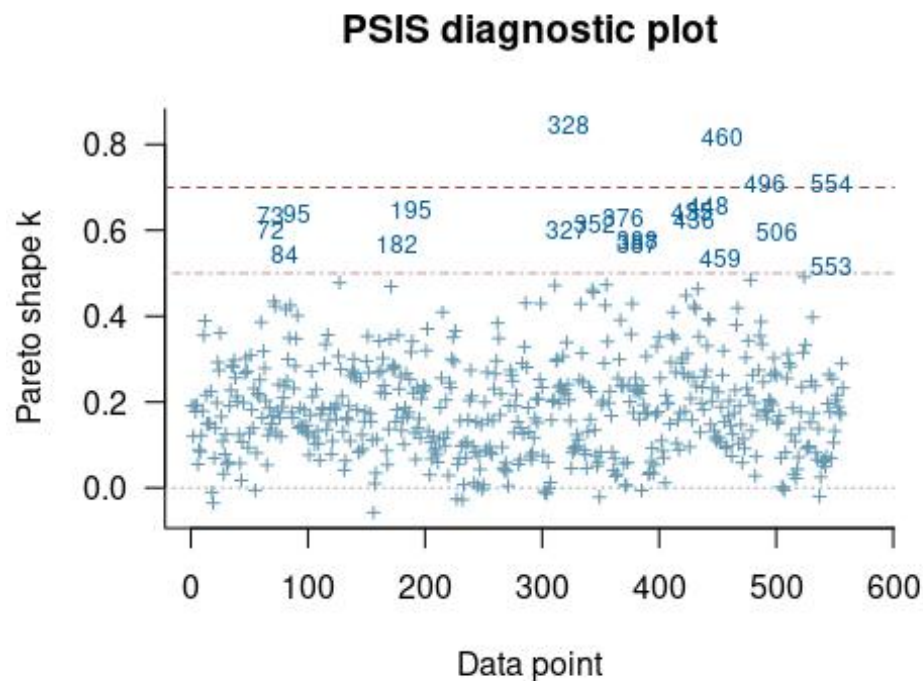


```
plot(loo2,label_points = TRUE)
```

**PSIS diagnostic plot**



```
plot(loo3,label_points = TRUE)
```



The Pareto  $k$  values of the first and second plots are mostly distributed around 0 and do not exceed the critical threshold of 0.7, which generally indicates that the LOO-CV estimation is reliable. This means that the model's predictions for most data points are robust.

The third graph shows that there are some data points whose Pareto  $k$  value exceeds the threshold of 0.7. These higher  $k$  values indicate that these particular observations may have had a destabilizing effect on the LOO-CV estimate. A value of  $k$  higher than 0.7 means that the LOO-CV estimate for the corresponding observation point may be too optimistic. The reason probably is that model can be overly complex, attempting to capture noise in the data rather than underlying signals, resulting in degraded predictive performance when certain data points are set aside for cross-validation.

(k)

#### Creating a dummy example plot

```
pareto_k_values <- loo(f3)$diagnostics$pareto_k
```

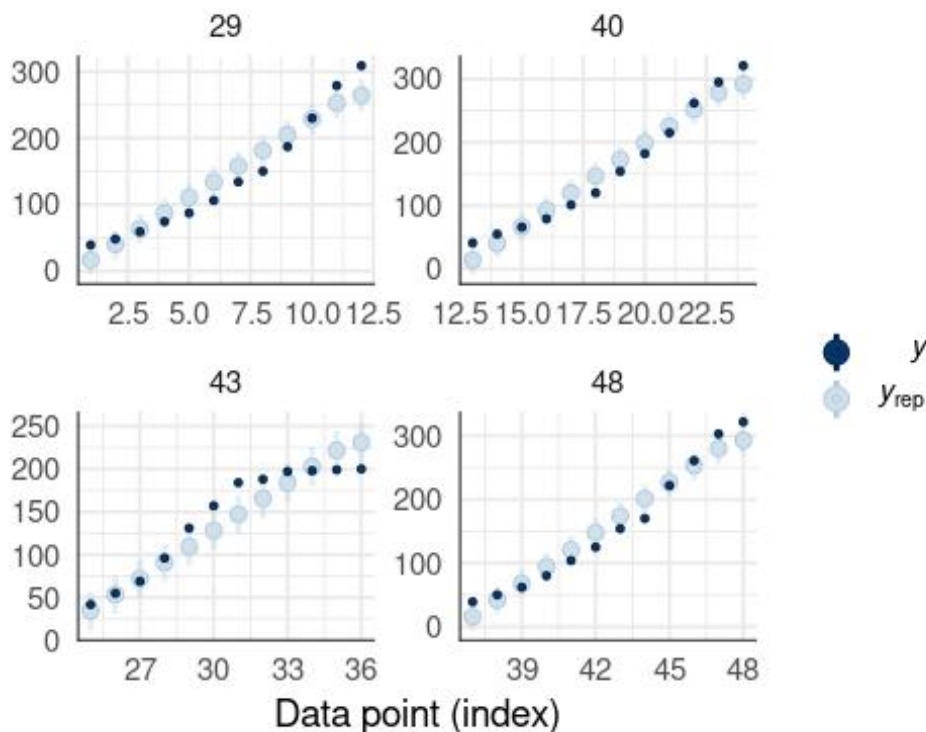
```
## Warning: Found 4 observations with a pareto_k > 0.7 in model 'f3'. It
  is
## recommended to set 'moment_match = TRUE' in order to perform moment m
atching
## for problematic observations.
```

```

# Identify chicken IDs with the highest Pareto k values
top_chicken_ids <- order(pareto_k_values, decreasing = TRUE)[1:4]
top_chicken_ids <- sort(top_chicken_ids)
top_chicken_indices <- ChickWeight$Chick[top_chicken_ids]
brms::pp_check(
  f3, type = "intervals_grouped", group = "Chick",
  newdata = ChickWeight |> dplyr::filter(Chick %in% top_chicken_indices)
)

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

```



```

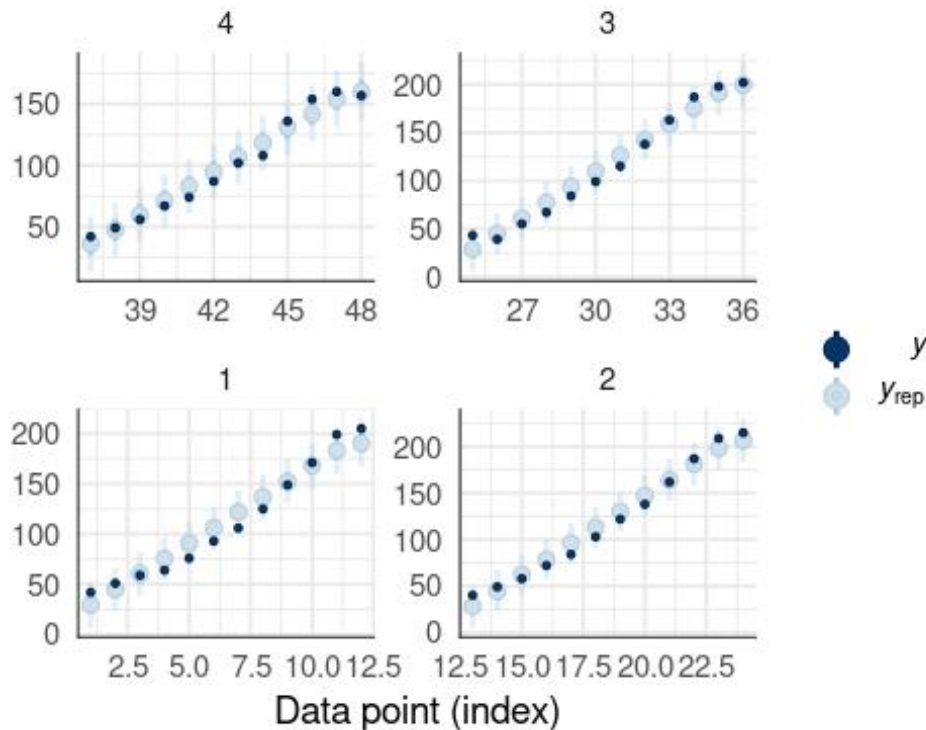
# Identify chicken IDs with the Lowest Pareto k values
low_pareto_chicken_indices <- numeric(0)
for (i in unique(ChickWeight$Chick)) {
  position <- which(ChickWeight$Chick == i)
  k_values <- pareto_k_values[position]
  has_low_k <- all(k_values <= 0.5)

  if (has_low_k) {
    low_pareto_chicken_indices <- c(low_pareto_chicken_indices, i)
  }
}
selected_low_k_chickens <- low_pareto_chicken_indices[1:4]
brms::pp_check(
  f3, type = "intervals_grouped", group = "Chick",
  newdata = ChickWeight |> dplyr::filter(Chick %in% selected_low_k_chick

```



```
ens)
)
## Using all posterior draws for ppc type 'intervals_grouped' by default.
```



Based on the graph, it appears that the model's predictive accuracy diminishes during later growth stages for chickens with high Pareto  $k$  values. Specifically, the predicted weights for these chickens at later time points are consistently overestimated when compared to the observed data. This discrepancy suggests that the model may not be capturing the growth dynamics accurately as the chickens mature, potentially indicating the need for model adjustments to better reflect the growth trajectory during this period.

## Model comparison using the RMSE

(I)

### rmse function implementation

```
# Compute RMSE or LOO-RMSE
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
  ))
}

rmse_f1 <- rmse(f1)
rmse_f2 <- rmse(f2)
rmse_f3 <- rmse(f3)

# compute LOO-RMSE
loo_rmse_f1 <- rmse(f1, use_loo = TRUE)

## Running PSIS to compute weights
## Warning: 'log_ratios' not specified. Can't compute k-hat diagnostic.
loo_rmse_f2 <- rmse(f2, use_loo = TRUE)

## Running PSIS to compute weights
## Warning: 'log_ratios' not specified. Can't compute k-hat diagnostic.
loo_rmse_f3 <- rmse(f3, use_loo = TRUE)

## Running PSIS to compute weights
## Warning: Some Pareto k diagnostic values are too high. See help('pare
to-k-diagnostic') for details.

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

list(rmse_f1 = rmse_f1, loo_rmse_f1 = loo_rmse_f1,
     rmse_f2 = rmse_f2, loo_rmse_f2 = loo_rmse_f2,
     rmse_f3 = rmse_f3, loo_rmse_f3 = loo_rmse_f3)

## $rmse_f1
## [1] 35.83317
##
## $loo_rmse_f1
## [1] 36.19547
##
## $rmse_f2
## [1] 34.80557
##
## $loo_rmse_f2
## [1] 35.12098
##
## $rmse_f3
## [1] 12.01964
##
## $loo_rmse_f3
## [1] 13.94057

```

Both RMSE and LOO-RMSE of model f3 are significantly lower than the corresponding values of models f1 and f2, which indicates that model f3 has better prediction performance in the case of the whole data set and cross-validation. But the LOO-RMSE of model f3 has the largest increase compared to RMSE, which may indicate that model f3's predictions for set aside observations may be less accurate than those on the entire dataset.

For each model (f1, f2, f3), the RMSE calculated based on all data is lower than the LOO-RMSE calculated based on leave-one cross-validation (LOO-CV). This is expected because LOO-RMSE is usually higher because in each reserve-one prediction, an observation is excluded and the model faces a slightly different data set in each prediction, which simulates the model's ability to predict new data.