Assignment 8

LOO-CV model comparison

anonymous

1 General information

This is the template for assignment 8. You can download 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 A hierarchical model for chicken weight time series

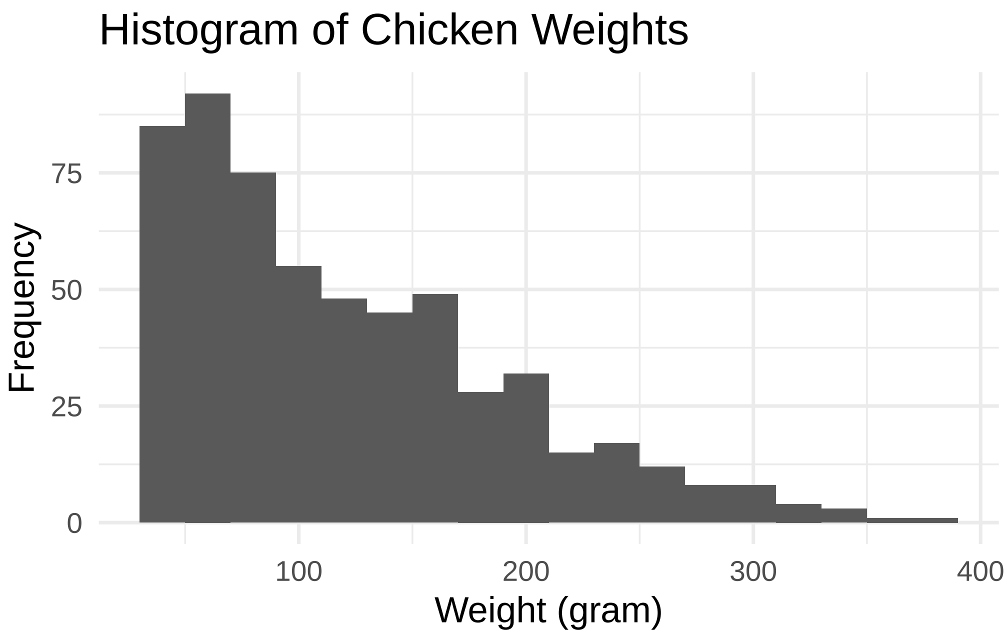
Referencing models: Linear regression model is referenced as f1 Log-normal linear regression model is referenced as f2 Hierarchical log-normal linear regression is meant as f3

2.1 Exploratory data analysis

2.2 (a)

```
data("ChickWeight")

# Create a histogram using ggplot
ggplot(ChickWeight, aes(x = weight,)) +
  geom_histogram(binwidth = 20) +
  labs(title = "Histogram of Chicken Weights",
       x = "Weight (gram)",
       y = "Frequency")
```



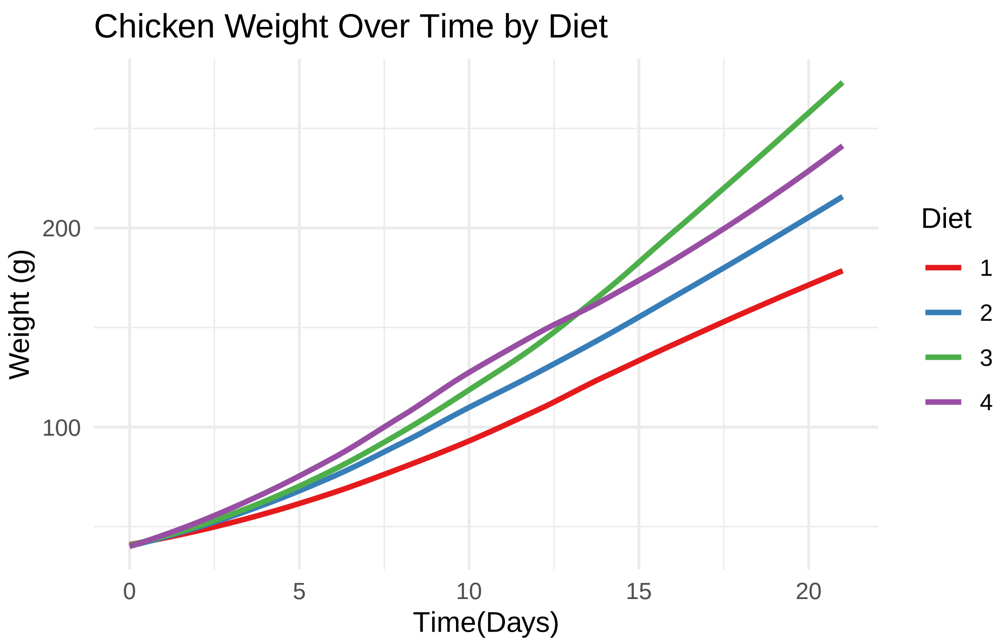
In the histogram of Chicken wights plot, one can see the distribution fo weights by the number of frequency each weight has been observed in the dataset. There are higher distribution of lower weights relatively to higher weights. This relationship seems to be dwindling. The reason for the relationship is probably because chicken weights are increasing for each day as the chickens are growing up. This can be confirmed with a Chicken weight over time.

2.3 (b)

```
# Plotting the line plot with ggplot
ggplot(data = ChickWeight, aes(x = Time, y = weight, group = Diet, color = Diet)) +

  geom_smooth(aes(color = as.factor(Diet)), method = 'loess', se = FALSE) +
  labs(title = "Chicken Weight Over Time by Diet",
       x = "Time(Days)",
       y = "Weight (g)") +
  theme_minimal() +
  scale_color_brewer(palette = "Set1")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



In the Chicken weight over time by diet graph one can see how each diet affected the weight over time. The diet 3 seems to be best for fastest growth. This graph has been smoothed for easier readability.

2.4 Linear regression

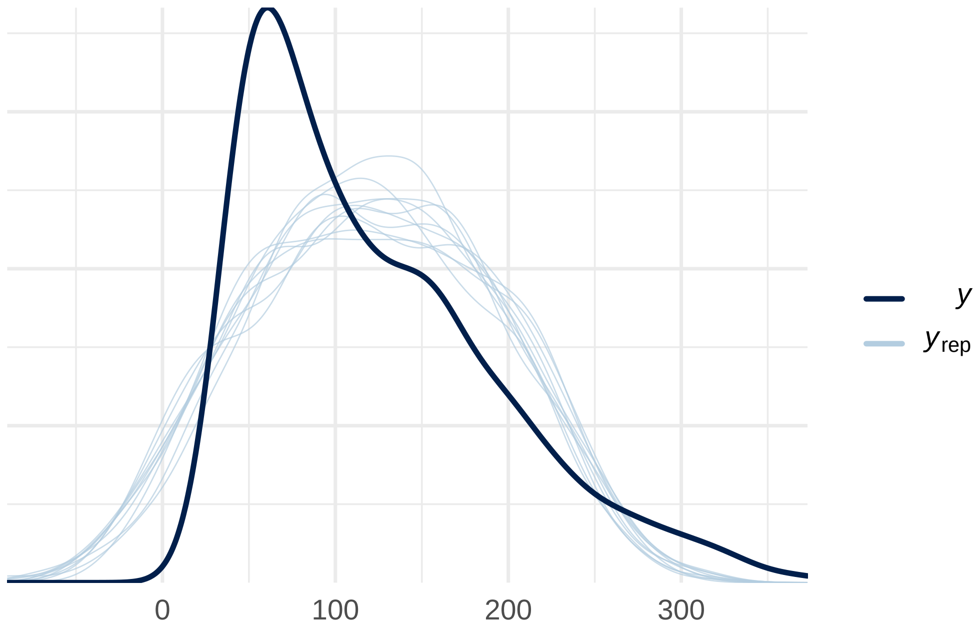
2.5 (c)

The priors are estimations based on observing Chicken weight over time by diet graph.

2.6 (d)

```
pp_check(f1)
```

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

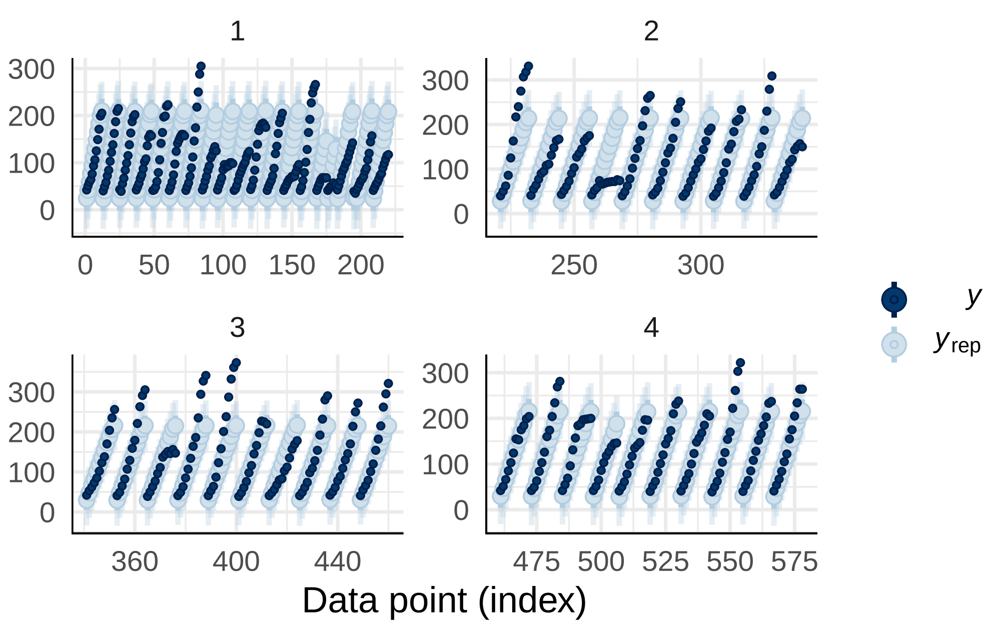


The plot compares observed data (y) to simulated data (y_red) from the posterior predictive distribution. From the observed data (y) and posterior predictive distribution (y_rep) are different which indicates model not being the best fit. Although this is only a visual tool one can not defintely say what is wrong.

2.7 (e)

```
pp_check(f1, type = "intervals_grouped", group = "Diet")
```

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



There seems again to be divergence between y_rep and y as the y wanders outside of the predictions.

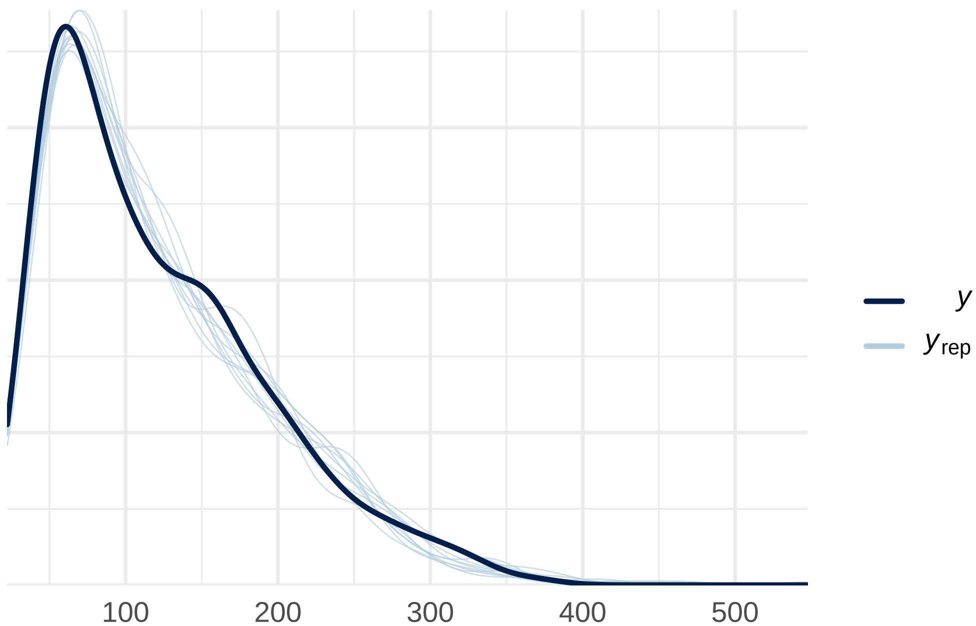
One way to improve the model is to choose another one and test if that would give results. There may out there be a model which is more suitable for out problem.

2.8 Log-normal linear regression

2.9 (f)

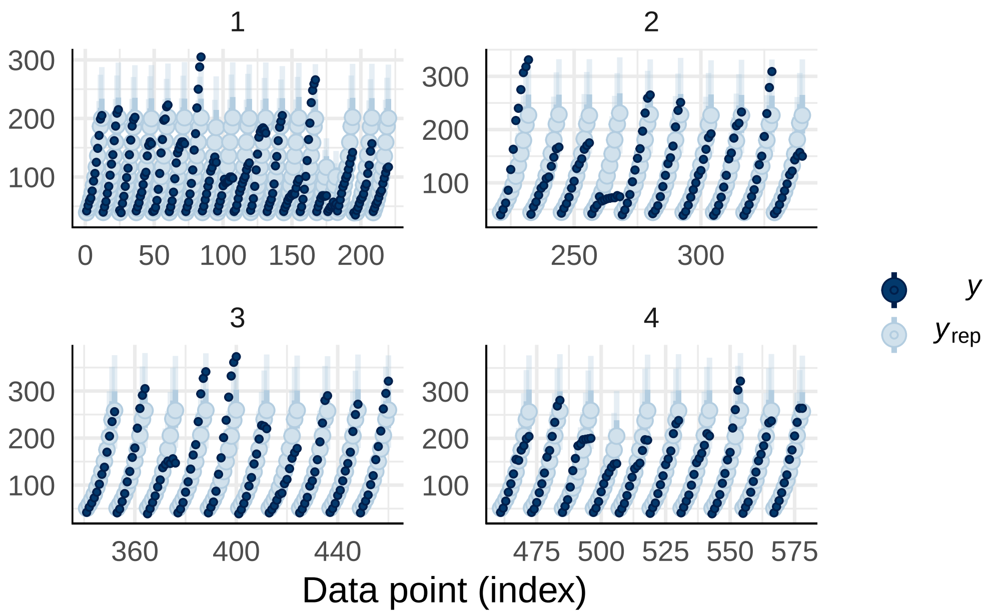
```
pp_check(f2)
```

```
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.
```



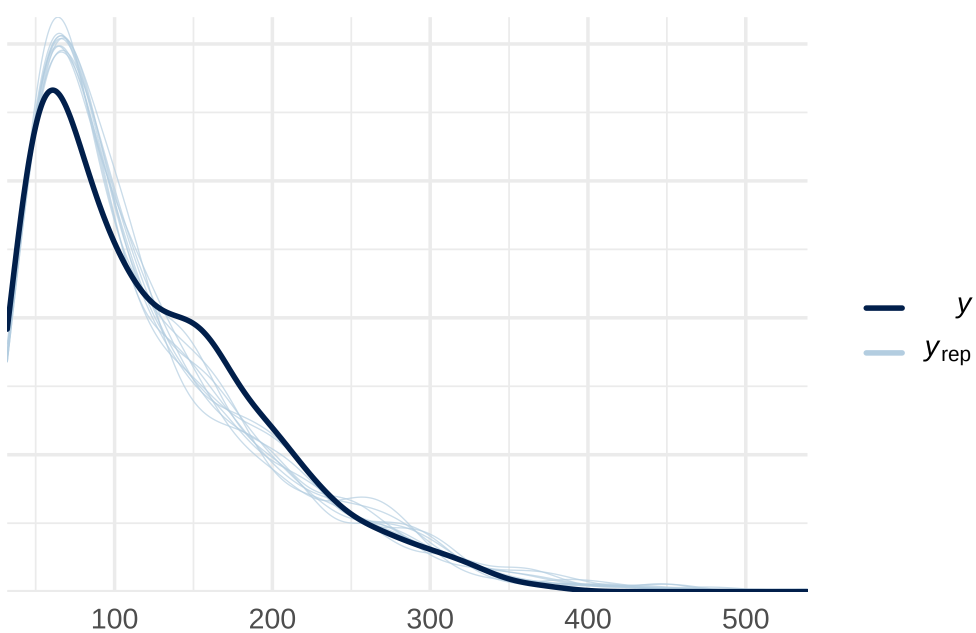
In the two plots above, it can be observed the model seems to fit better than in the normal linear regression. There is some variation in the predictions but considerbly less than in the linear regression model. There are still some divergence between predictions and observations.

2.10 Hierarchical log-normal linear regression

2.11 (g)

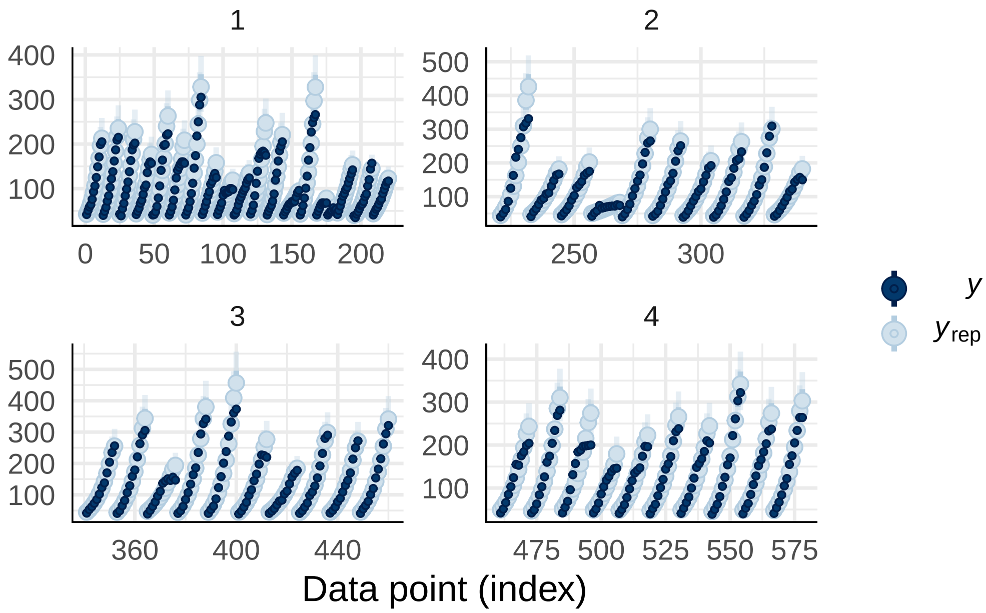
```
pp_check(f3)
```

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



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

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



In the two plots above, it can be observed the model seems to fit better than in the normal linear regression and log normal linear regression.

There are some marginal deviations between the observed and predicted data in some groups but more analysis should be done to find the root cause. In this case one starts getting diminishing returns for improved models and more complex model may lead to overfitting.. In the case one would want to try improving the model. Firstly, Reviewing and potentially adjusting the priors to better reflect the known constraints or beliefs about the data could also improve the model. Secondly, Investigating the influence of potential outliers on the model's predictions.

2.12 (h)

All models have reached an Rhat value of 1. ESS have all been over the threshold value of 400. Thus all three models (f1, f2 and f3) have reached convergence.

2.13 Model comparison using the ELPD

2.14 (i)

```r
# Useful functions: loo, loo_compare
loo(f1,f2,f3)
```

```
Warning: Found 2 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.
```

```
Output of model 'f1':

Computed from 4000 by 578 log-likelihood matrix

         Estimate   SE
elpd_loo  -2925.1 26.8
p_loo         4.8  0.7
looic      5850.2 53.7
------
Monte Carlo SE of elpd_loo is 0.0.

All Pareto k estimates are good (k < 0.5).
See help('pareto-k-diagnostic') for details.

Output of model 'f2':

Computed from 4000 by 578 log-likelihood matrix

         Estimate   SE
elpd_loo  -2649.4 30.8
p_loo         7.1  1.1
looic      5298.8 61.7
------
Monte Carlo SE of elpd_loo is 0.0.

All Pareto k estimates are good (k < 0.5).
See help('pareto-k-diagnostic') for details.

Output of model 'f3':

Computed from 4000 by 578 log-likelihood matrix

         Estimate   SE
elpd_loo  -2254.8 27.2
p_loo        78.8  6.8
looic      4509.6 54.3
------
Monte Carlo SE of elpd_loo is NA.

Pareto k diagnostic values:
                         Count Pct.    Min. n_eff
(-Inf, 0.5]   (good)     556   96.2%   438
 (0.5, 0.7]   (ok)        20    3.5%   111
   (0.7, 1]   (bad)        2    0.3%   22
   (1, Inf)   (very bad)   0    0.0%   <NA>
See help('pareto-k-diagnostic') for details.

Model comparisons:
   elpd_diff se_diff
f3    0.0       0.0
f2 -394.6      30.2
f1 -670.3      32.8
```

```r
loo_compare(loo(f1),loo(f2), loo(f3) )
```

```
Warning: Found 2 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.
```

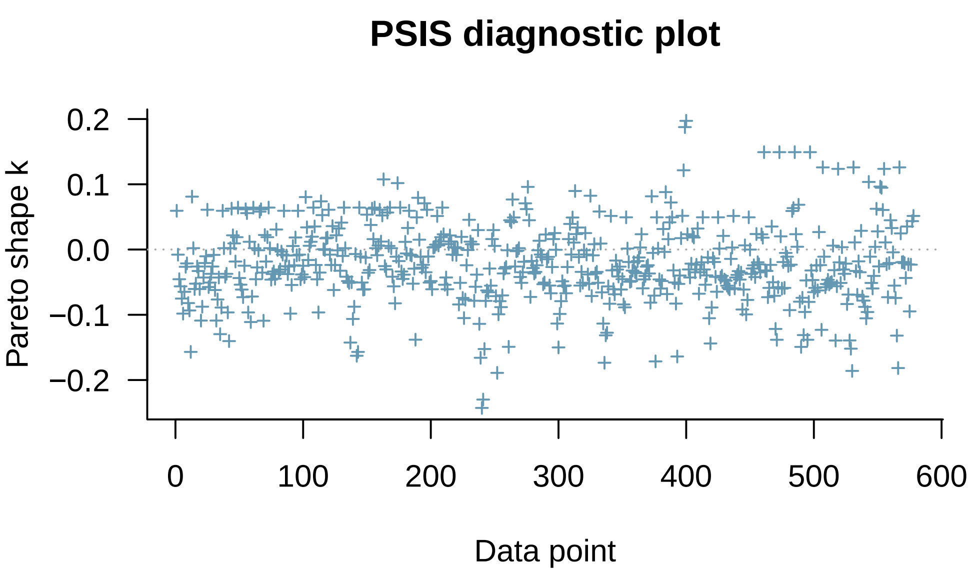
```
   elpd_diff se_diff
f3    0.0       0.0
f2 -394.6      30.2
f1 -670.3      32.8
```

Model f3 has the best predictive performance based on the results since it has the highest (least negative) elpd_loo value.

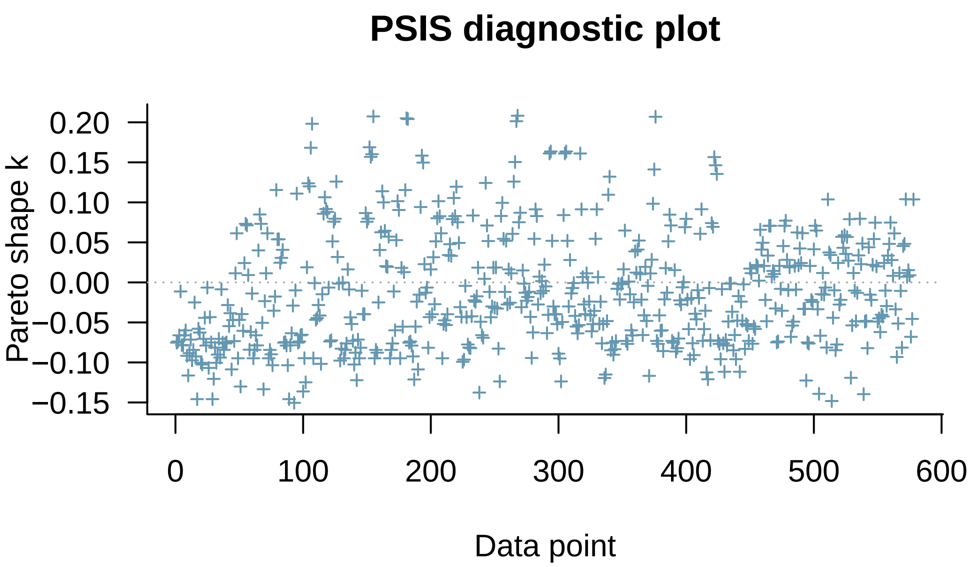
In this case, the standard errors (elpd_diff) are relatively small compared to the magnitude of the differences in elpd_loo values. This means that the uncertainty in the estimates does not significantly influence the decision of which model is best. There is enough evidence to confidently state that f3 outperforms f1 and f2 in terms of predictive accuracy.

2.15 (j)

```r
plot(loo(f1), label_points = TRUE)
```

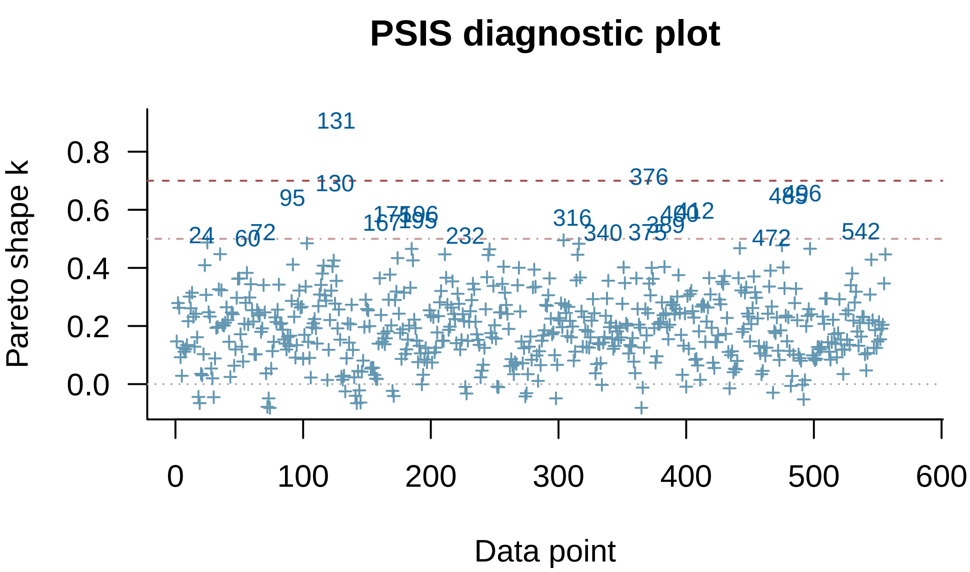



```r
plot(loo(f2), label_points = TRUE)
```




```r
plot(loo(f3), label_points = TRUE)
```

```
Warning: Found 2 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.
```



In a reliable model, most of the Pareto k values should ideally be less than 0.5. Values between 0.5 and 0.7 may be acceptable but indicate that the results should be interpreted with caution, and values above 0.7 suggest that the approximation may be unreliable for the corresponding data points.

In models f1 and f2 all values are under 0.5 which is ideal. In the model f3 most of the points are under 0.5 a small amount between 0.5 and 0.7 and only 1 value over 0.7. Thus as model f3 has presence of these high Pareto k values does not invalidate the entire LOO-CV analysis, but it does suggest that the model's predictive performance might be overestimated.

The model f2 appears to provide a more reliable LOO-CV estimation than f3 based on the PSIS diagnostic plots. Even though f3 may have shown better predictive performance in terms of raw elpd_loo scores, the reliability of these scores is questionable because of the high number of high k values.

2.16 (k)

2.17 Model comparison using the RMSE

2.18 (l)

```r
# 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)
```

```
[1] 37.89528
```

```r
rmse(f2)
```

```
[1] 34.77954
```

```r
rmse(f3)
```

```
[1] 15.80801
```

f1 = 38 f2 = 34 f3 = 16

RMSE measures the average prediction error using training data, while LOO-RMSE estimates error through cross-validation, indicating generalization to new data. LOO-RMSE typically exceeds RMSE, reflecting the model's ability to handle unseen data versus fitting to the training set.

LOO-RMSE can be expected to be higher than RMSE because LOO-RMSE assesses model performance on data not used during training, thus incorporating the model's ability to generalize. RMSE, on the other hand, might be optimistic as it measures error on the same data the model has seen

AI was not used.