CS-E5710 Bayesian data analysis D

Final report: Predicting number of yellow cards in a football match

Rongjun Ma, Aalto University 727794 Dusan Vukic, Aalto University 1023756

Contents

Introduction

Motivation

Predicting football results is a very popular topic. Most often, predictions are dealing with the final outcome of the matches (win / lose / draw) or the actual results (number of goals scored).

We were, however, interested in one of the auxiliary outcomes of the football match - number of yellow cards. Referees are using yellow cards as a way to caution players for various kinds of offences. Most often, yellow cards are shown for reckless or intentional fouls. They can also be shown for dissent, unsporting behaviour or breaking any of the rules of the game.

Referees can also show red cards, for more serious fouls, committed using excessive force, and for serious misconduct (red cards are not part of this project).

Decision on whether an action deserves a card penalty (and if yes, is it a yellow or a red card) is following strict rules. But still, it is up to the referee to make a final judgement, based on his view of the situation.

Idea

We wanted to check if occurrences of yellow cards are generally the same for every team, or whether some teams are more prone to getting yellow cards than the others. Is there a significant difference in the number of yellow cards teams get during the match and can we make any prediction of this outcome?

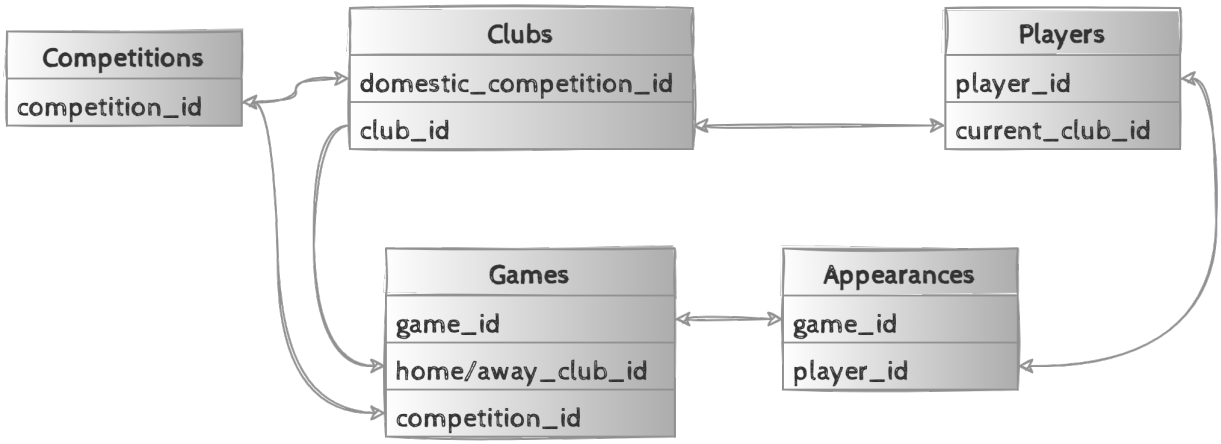
We believe that the number of yellow cards should be inversely correlated to team quality. Teams with lower skill level usually need to compensate by playing more aggressively, resulting in more risky fouls when trying to stop the attack.

On the other hand, 'aggressiveness' can be a part of the team's tactics - a conscious decision by a team manager to instruct his players to make more or less fouls. Also, some players may be inherently more aggressive, keeping their teams' card score at higher levels.

Dataset

The source

Data for the project is obtained from data.world website (https://data.world/dcereijo/player-scores). It is scraped from Transfermarkt.com, a popular German website that collects statistics about football teams, matches and players.
The full dataset consists of several connected tables, as shown on this image.



For the project, we aggregated the data into a smaller table, keeping only club_id, game_id and sum(yellow_cards). We decided to focus on only one competition - English Premier League (EPL) and on only one season - 2019.
EPL is the top level league in the English football system, and it consists of 20 teams. During one season, each team plays two matches against each opponent (total 38 matches). This means that our final dataset has 760 datapoints. For simplicity, teams and matches are indexed in the dataset, so that teams are represented by values (1-20) and matches by values (1-38). Outcomes (number of yellow cards in a match) are integers - in this case, having values (0-7).

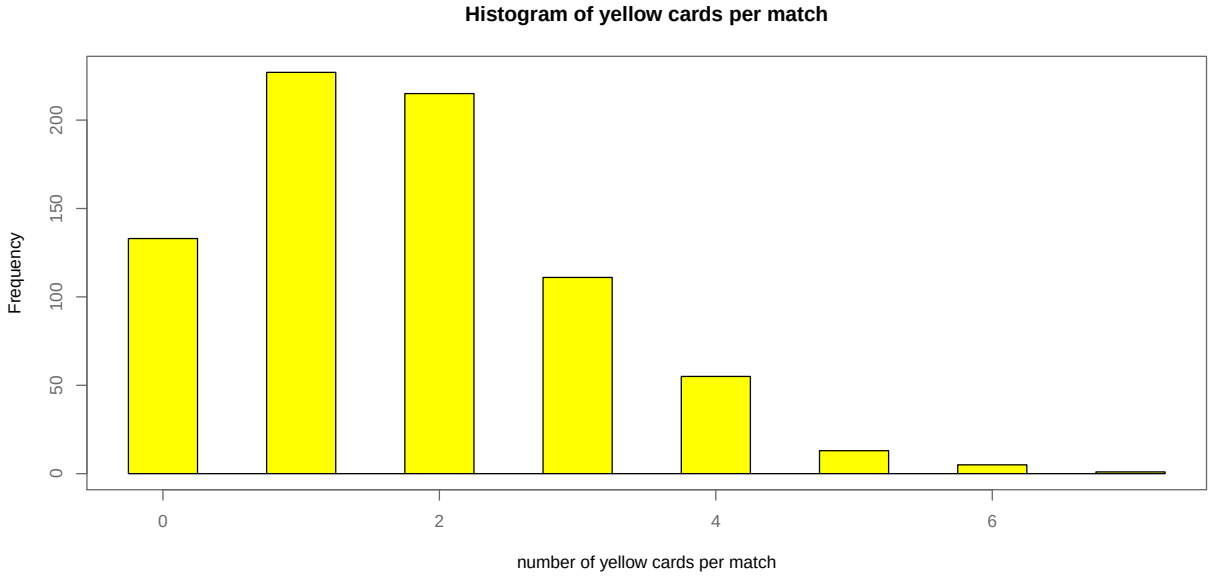
Exploratory analysis

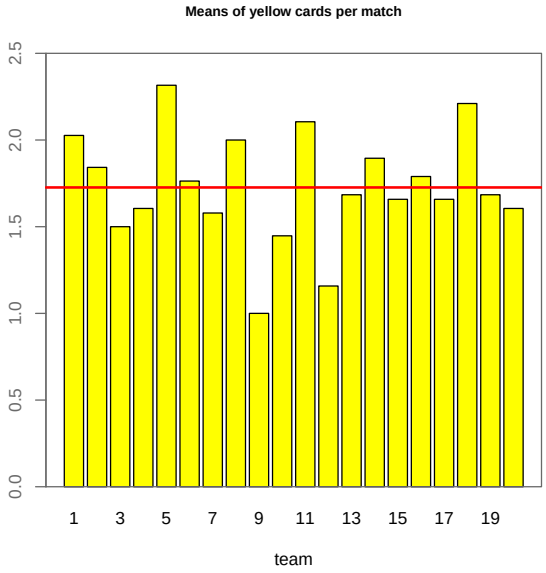
```
data <- read.csv('datasets/yellow_cards.csv')
head(data)
```

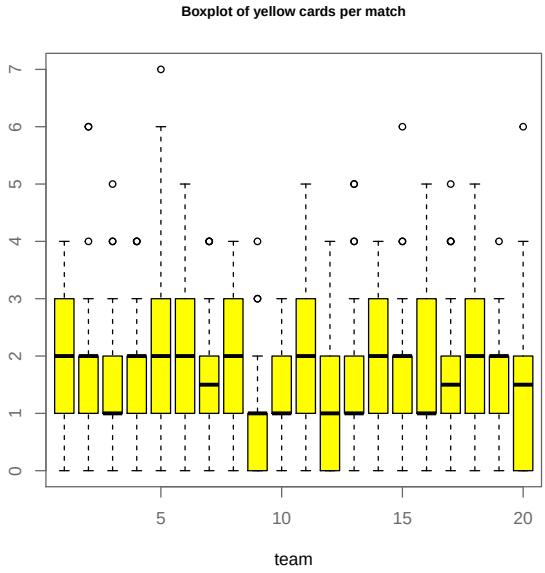
```
##         team_name team match is_home opponent_yellow_cards yellow_cards
## 1 Afc Bournemouth    1     1       1                     1            2
## 2 Afc Bournemouth    1     2       0                     1            3
## 3 Afc Bournemouth    1     3       1                     4            0
## 4 Afc Bournemouth    1     4       1                     3            1
## 5 Afc Bournemouth    1     5       0                     1            0
## 6 Afc Bournemouth    1     6       1                     1            2
```

Overall, the most common number of yellow cards per match is 1. Min value is 0, max value is 7.
Mean values per team range from 1.0 to 2.3, and overall mean is 1.7.
For most of the teams, IQR range is 1-3, and values higher than 4-5 are outliers.
Team 5 is extreme in the high end, with upper quarter going up to 6, and value 7 as an outlier.

Team 9 is extreme in the low end, with upper quarter as low as 2, and values 3 and 4 as outliers.







Modeling

Models and priors

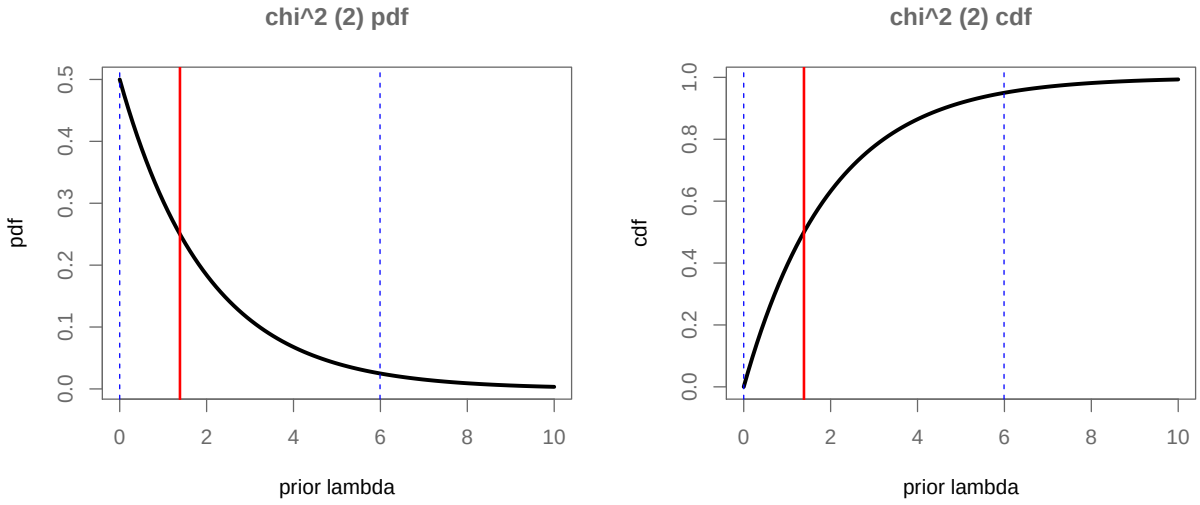
By analyzing distribution of the outcomes, we concluded that Poisson distribution would be the best fit.

Model 1: separate model

Each team has it's own λ parameter, and they are all using the same prior.

$$y \sim poisson(\lambda)$$
$$\lambda_i \sim \chi^2(2)$$

Selection of prior for parameter λ in based on distribution of observed values. It needs to be positive, mean should be between 1 and 2 and most of the density should be in the 0-7 interval.
$\lambda \sim \chi^2(2)$ has mean at 1.4 and 95% interval is (0,6). So it roughly satisfies all these conditions, and seems to be a valid choice for a prior.

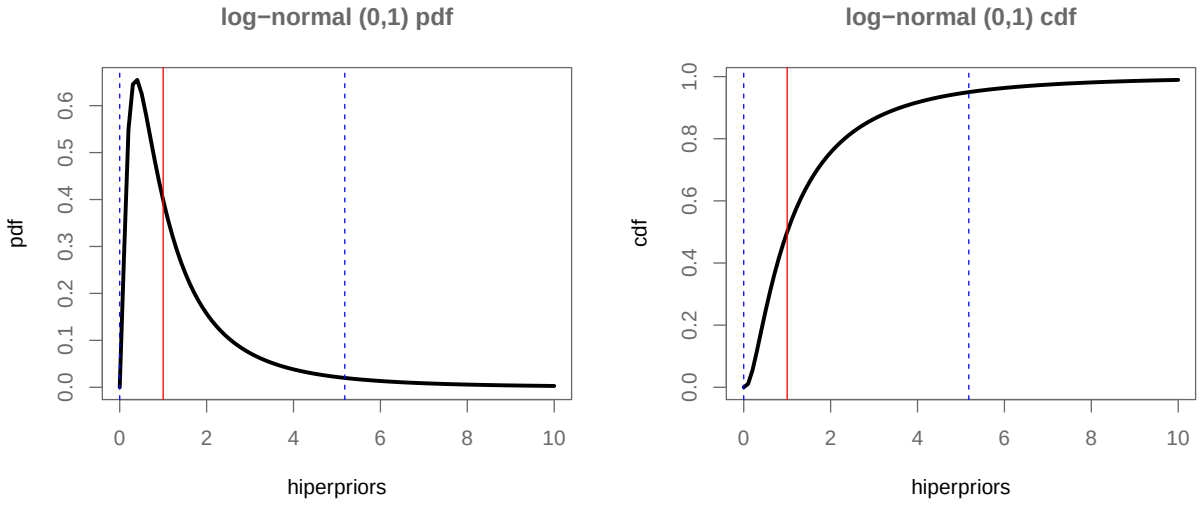


Model 2: hierarchical model

Parameter λ for each team is now coming from $lognormal(\mu, \sigma)$ prior (log-normal instead of normal because λ needs to be positive), where μ and σ are hyperpriors. We are keeping μ the same for all teams, while σ stays separate.

$$y \sim poisson(\lambda)$$
$$\lambda_i \sim lognormal(\mu, \sigma_i)$$
$$\mu \sim lognormal(0, 1)$$
$$\sigma_i \sim lognormal(0, 1)$$

Hyperpriors $\mu \sim lognormal(0,1)$ and $\sigma \sim lognormal(0,1)$ have their means at 1 and 95% interval at (0,5). So, it roughly fits into our expectations for λ parameter, as is already shown in priors analysis for model 1.



Model 3: hierarchical model with one predictor

As a predictor, we are now adding number of opponents' yellow cards in the same match. This reasoning is coming from the assumption that in a 'heated' match, both teams will have an increased probability of getting a yellow card. We are now using log-Poisson distribution of parameter $\theta = log\lambda$.

$$y \sim poissonlog(\theta)$$

$$\theta_i = x \cdot \beta_i + \alpha_i$$

$$\alpha_i = N(0,1)$$

$$\beta_i = N(0,1)$$

x - predictor (number of opponents' yellow cards)

θ is calculated using linear model for predictor values: $\theta = x \cdot \beta + \alpha$. So in this case, we need to select priors for intercept α and slope β parameters. Without any detailed investigation, we decided to use weekly informative priors $\alpha = N(0,1)$ and $\beta = N(0,1)$.

Stan code

Stan code for **model 1**:

```
## //separate model without predictors
## data {
##   int<lower=0> J;       //number of teams
##   int<lower=0> N;       //number of matches
##   int<lower=0> y[J,N];  //number of yellow cards
## }
## parameters {
##   real<lower=0> lambda[J];
## }
## model {
##   for (j in 1:J){
##     lambda[j] ~ chi_square(2);
##     y[j,] ~ poisson(lambda[j]);
##   }
## }
## generated quantities {
##   int<lower=0> ypred[J];
##   int y_rep[J,N];
##   real log_lik[J,N];
##   for (j in 1:J){
##     ypred[j] = poisson_rng(lambda[j]);
##     for (n in 1:N){
##       y_rep[j,n] = poisson_rng(lambda[j]);
##       log_lik[j,n] = poisson_lpmf(y[j,n] | lambda[j]);
##     }
##   }
## }
```

Stan code for **model 2**:

```
## //hierarchical model without predictors
## data {
##   int<lower=0> N;       //number of matches
##   int<lower=0> J;       //number of teams
##   int<lower=0> y[J,N];  //number of yellow cards
## }
## parameters {
##   real<lower=0> sigma[J];
##   real<lower=0> mu;
##   real<lower=0> lambda[J];
## }
## model {
##   mu ~ lognormal(0,1);
##   for (j in 1:J){
##     sigma[j] ~ lognormal(0,1);
##     lambda[j] ~ lognormal(mu,sigma[j]);
##     y[j,] ~ poisson(lambda[j]);
##   }
## }
```

```
## generated quantities {
##   int<lower=0> ypred[J];
##   int y_rep[J,N];
##   real log_lik[J,N];
##   for (j in 1:J) {
##     ypred[j] = poisson_rng(lambda[j]);
##     for (n in 1:N) {
##       y_rep[j,n] = poisson_rng(lambda[j]);
##       log_lik[j,n] = poisson_lpmf(y[j,n] | lambda[j]);
##     }
##   }
## }
```

Stan code for **model 3**:

```
## //hierarchical model with predictor - opponents' yellow cards
## data {
##   int<lower=0> N;       //number of matches
##   int<lower=0> J;       //number of teams
##   vector[N] x[J];       //number of yellow cards of opponent team
##   int<lower=0> y[J,N];  //number of yellow cards
## }
## parameters {
##   real alpha[J];
##   real beta[J];
## }
## transformed parameters {
##   real theta[J];        //log(lambda)
##   real lambda[J];
##   for (j in 1:J)
##     for (n in 1:N){
##       theta[j] = x[j,n] * beta[j] + alpha[j];
##       lambda[j] = exp(theta[j]);
##     }
## }
## model {
##   alpha ~ normal(0,1);
##   beta ~ normal(0,1);
##   for (j in 1:J) {
##     y[j,] ~ poisson_log(theta[j]);
##   }
## }
## generated quantities {
##   int<lower=0> ypred[J];
##   int y_rep[J,N];
##   real log_lik[J,N];
##   for (j in 1:J) {
##     ypred[j] = poisson_log_rng(theta[j]);
##     for (n in 1:N){
##       y_rep[j,n] = poisson_log_rng(theta[j]);
##       log_lik[j,n] = poisson_log_lpmf(y[j,n] | theta[j]);
##     }
##   }
## }
```

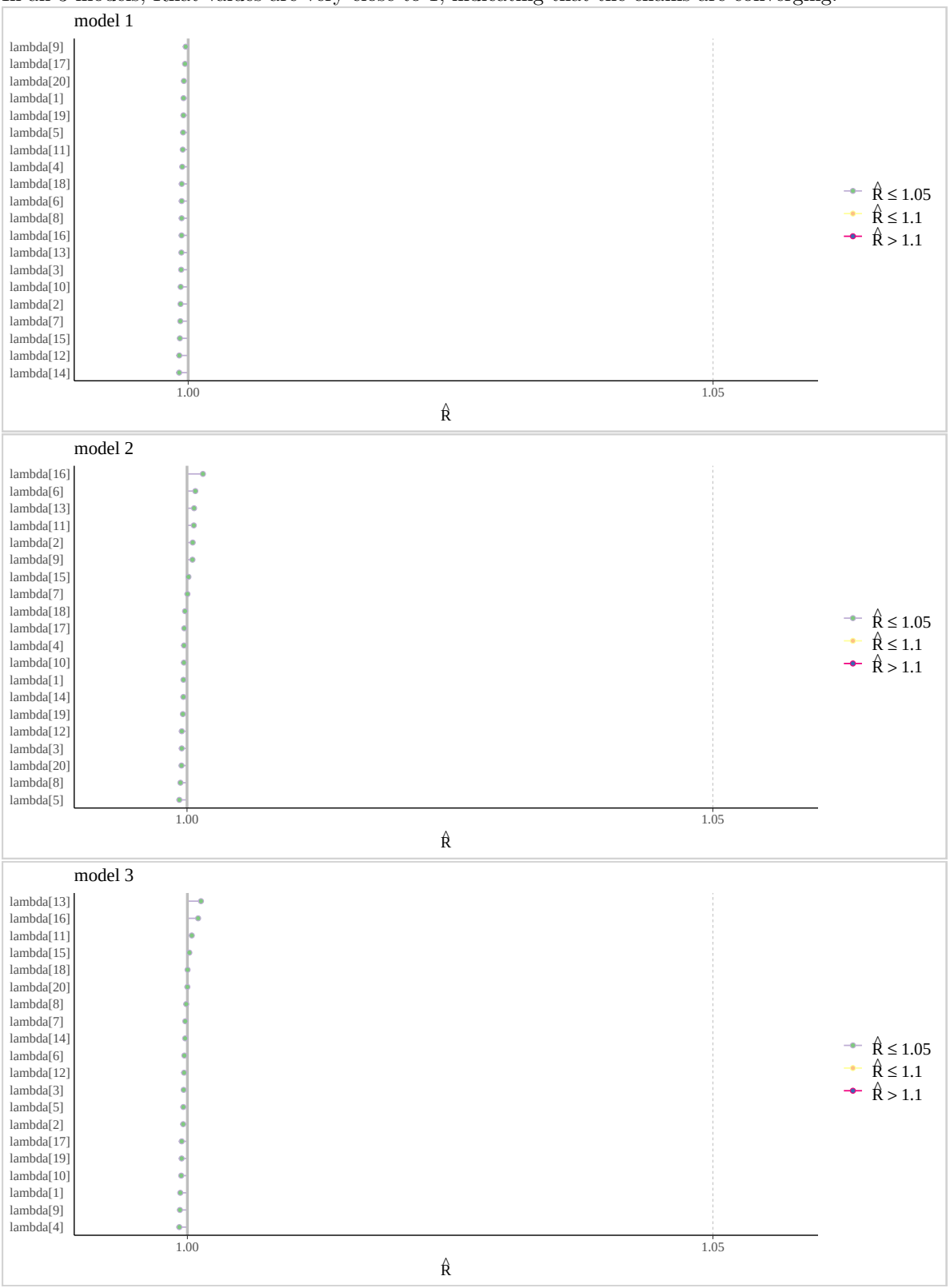
Stan code execution

```
stan_data <- list(y=split(data$yellow_cards, data$team),
                  x=split(data$opponent_yellow_cards, data$team),
                  N=38,  #number of matches for each team
                  J=20  # number of teams
                  )
```

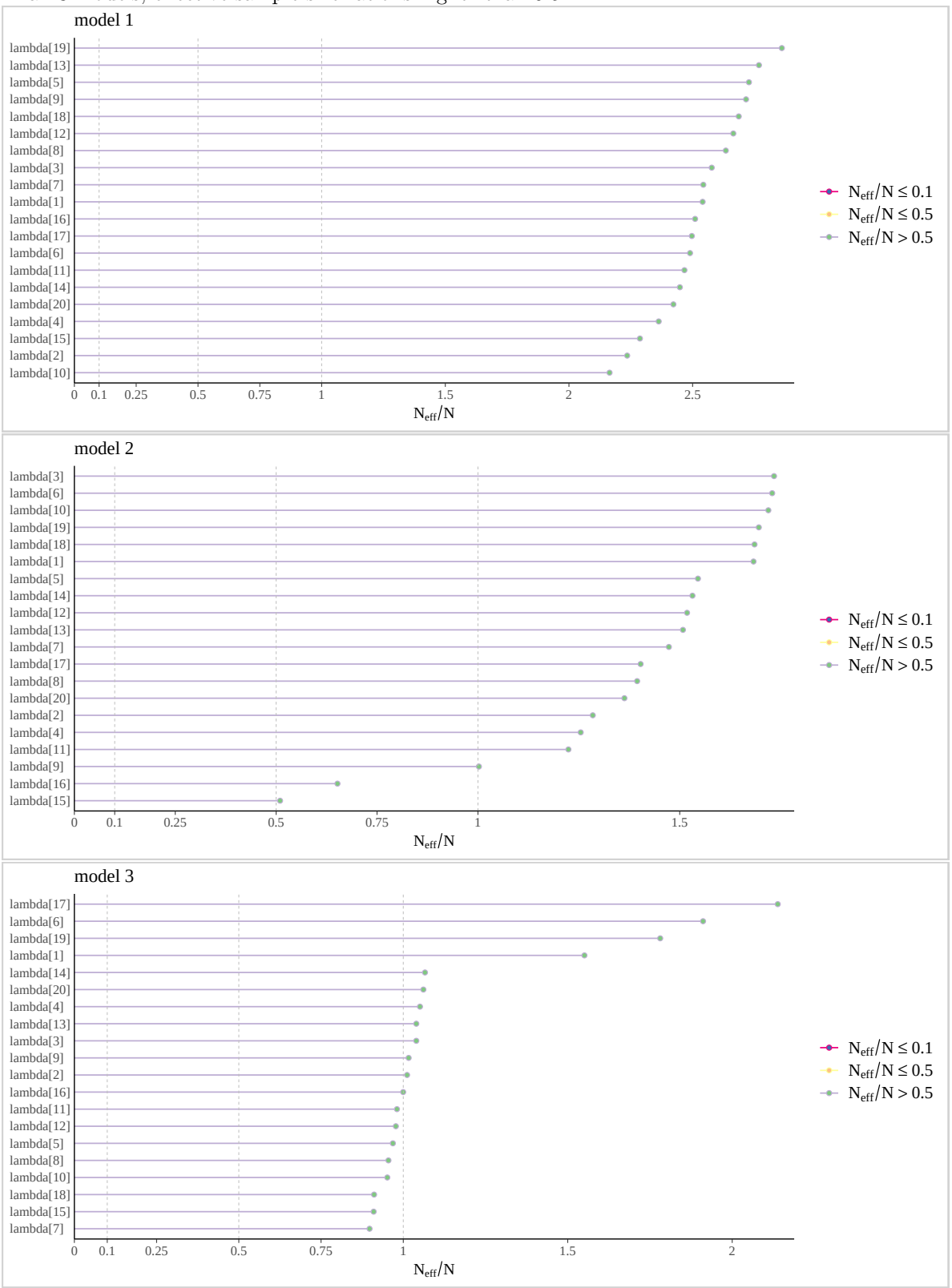
```
model_1 <- stan(file = "models/model_1.stan", data = stan_data, seed=seed)
model_2 <- stan(file = "models/model_2.stan", data = stan_data, seed=seed)
model_3 <- stan(file = "models/model_3.stan", data = stan_data, seed=seed)
```

Convergence diagnostics

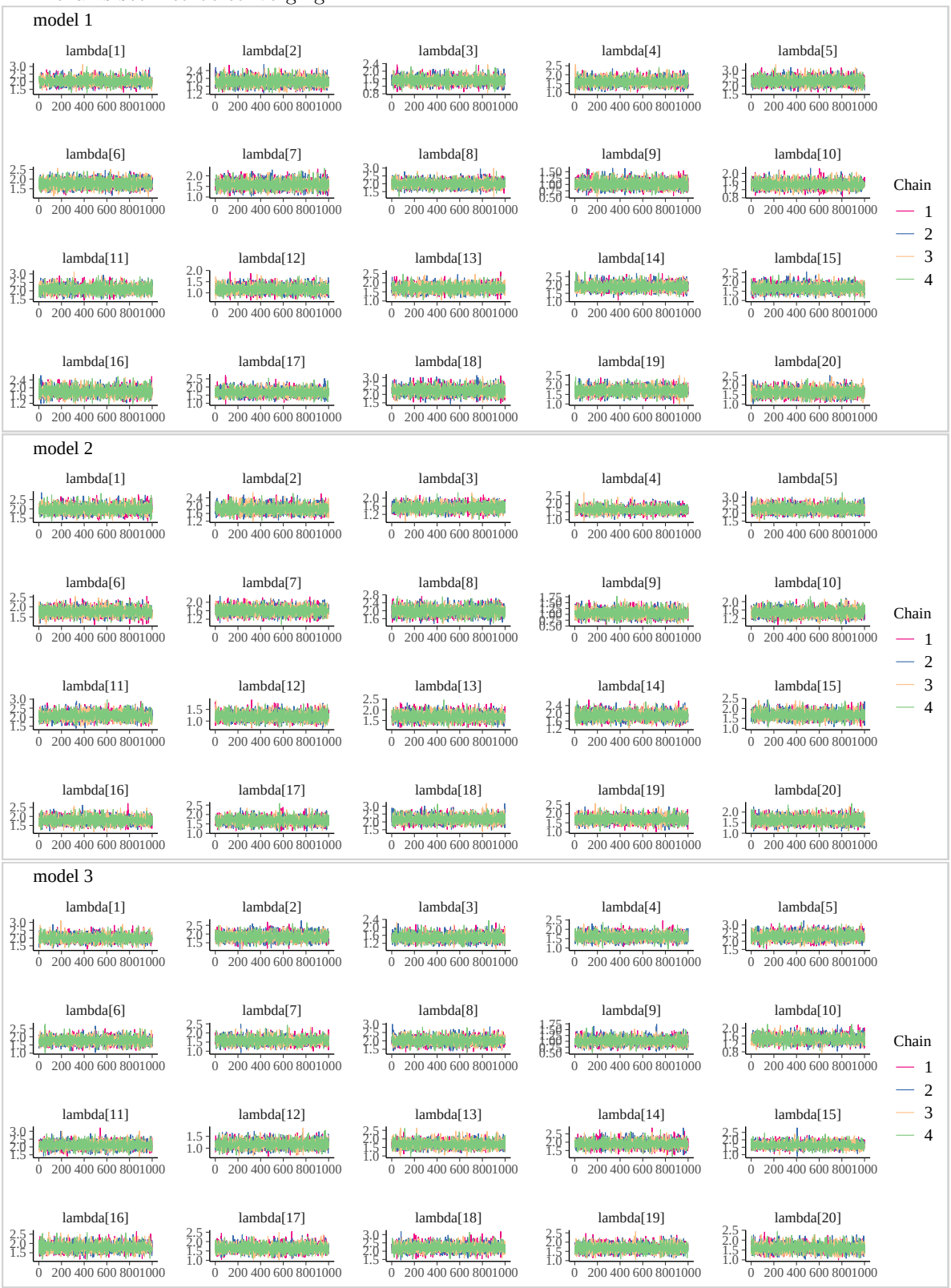
- In all 3 models, Rhat values are very close to 1, indicating that the chains are converging.



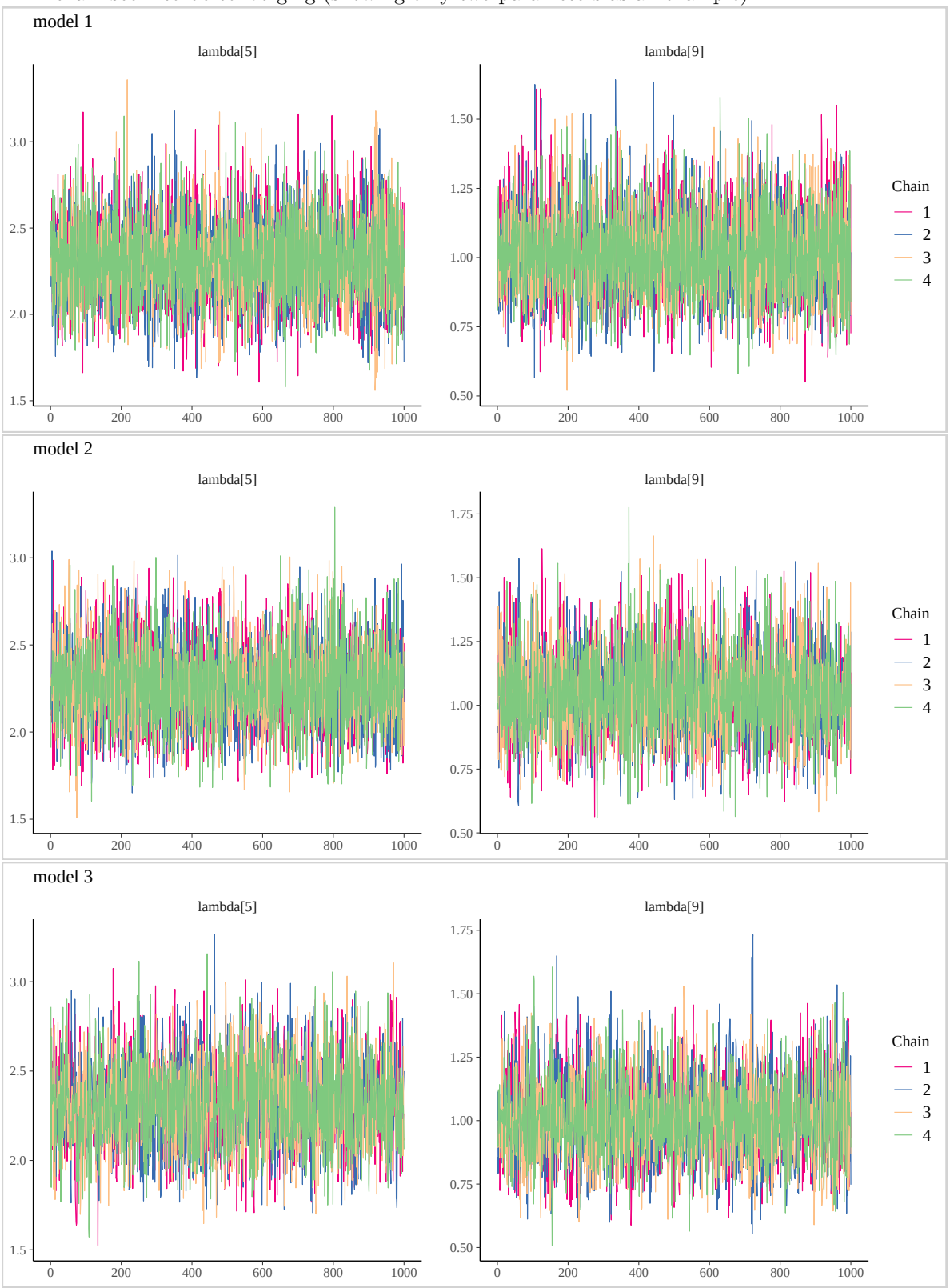
- In all 3 models, effective sample size ratio is higher than 0.5.



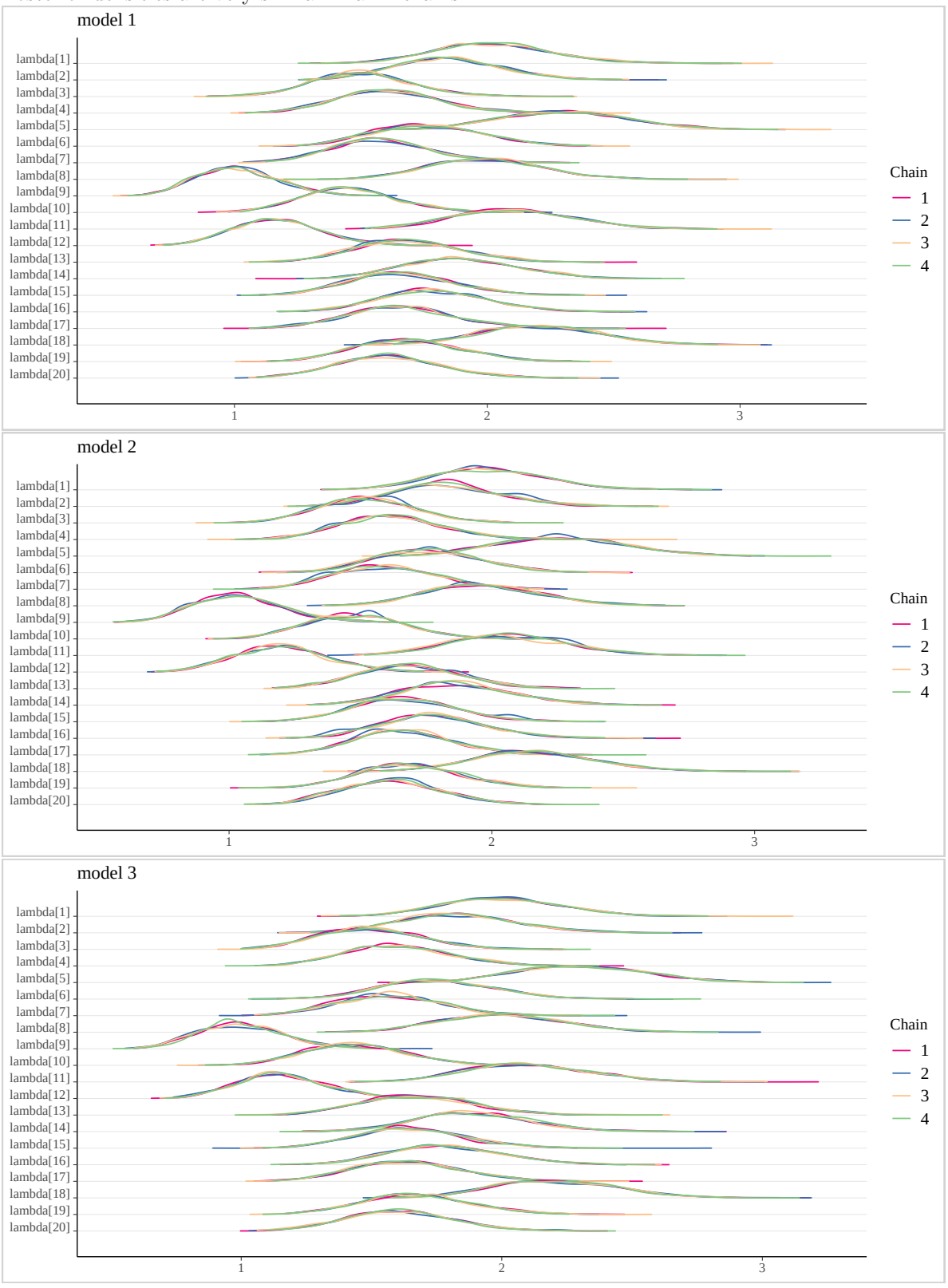
- All 4 chains seem to be converging.



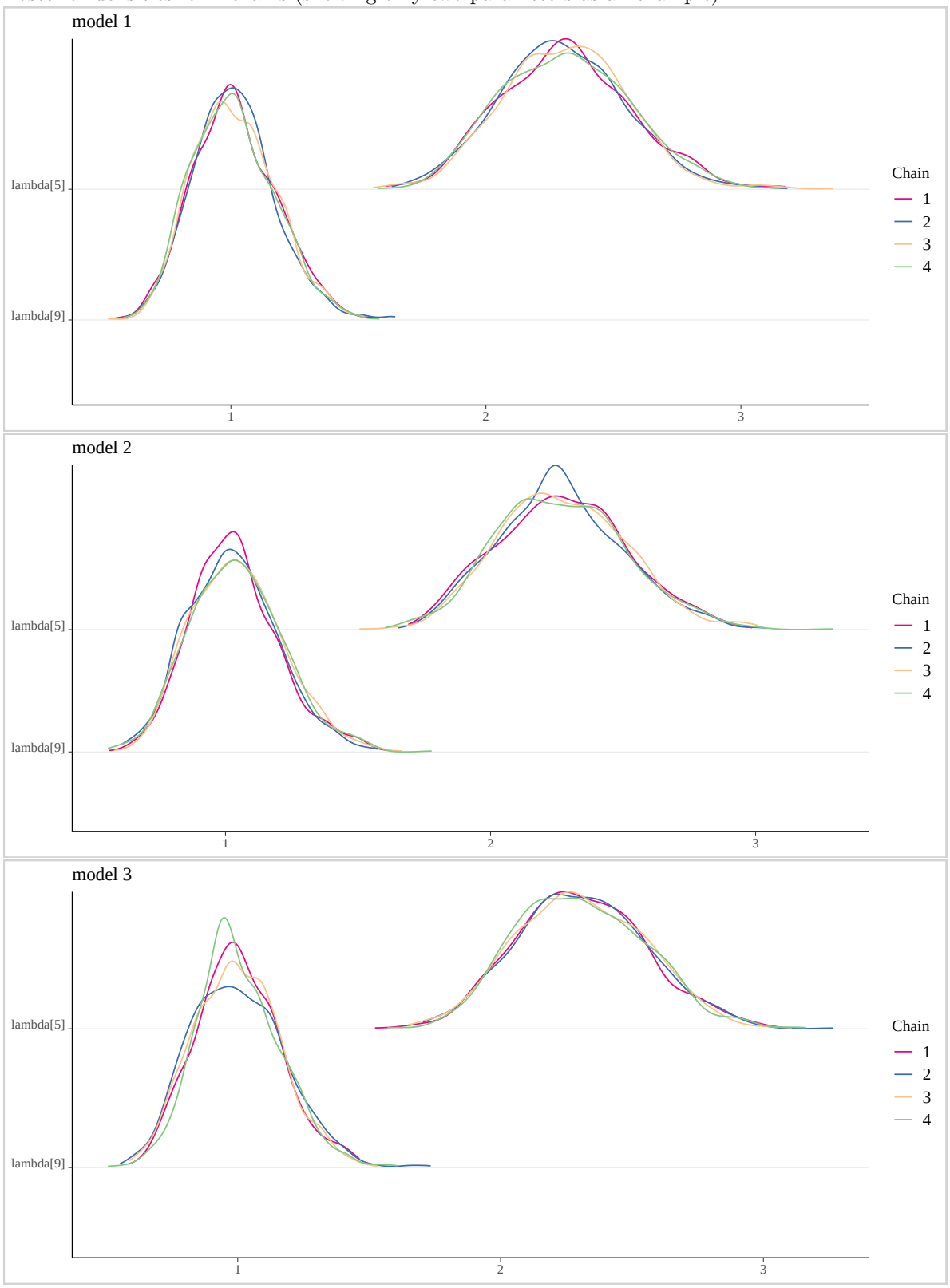
- All 4 chain seem to be converging (showing only two parameters as an example).



- Posterior densities are very similar in all 4 chains.

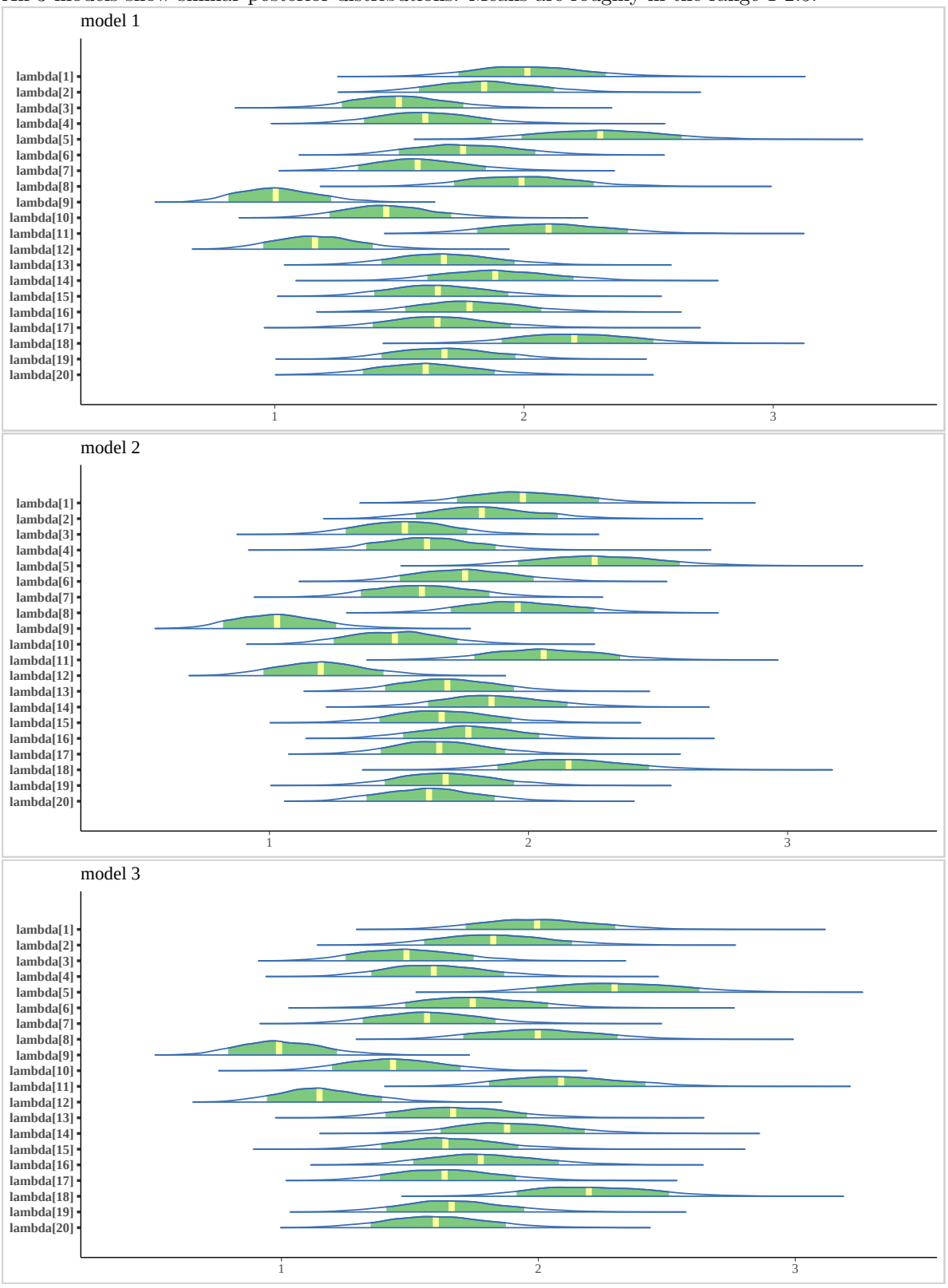


- Posterior densities for 4 chains (showing only two parameters as an example).

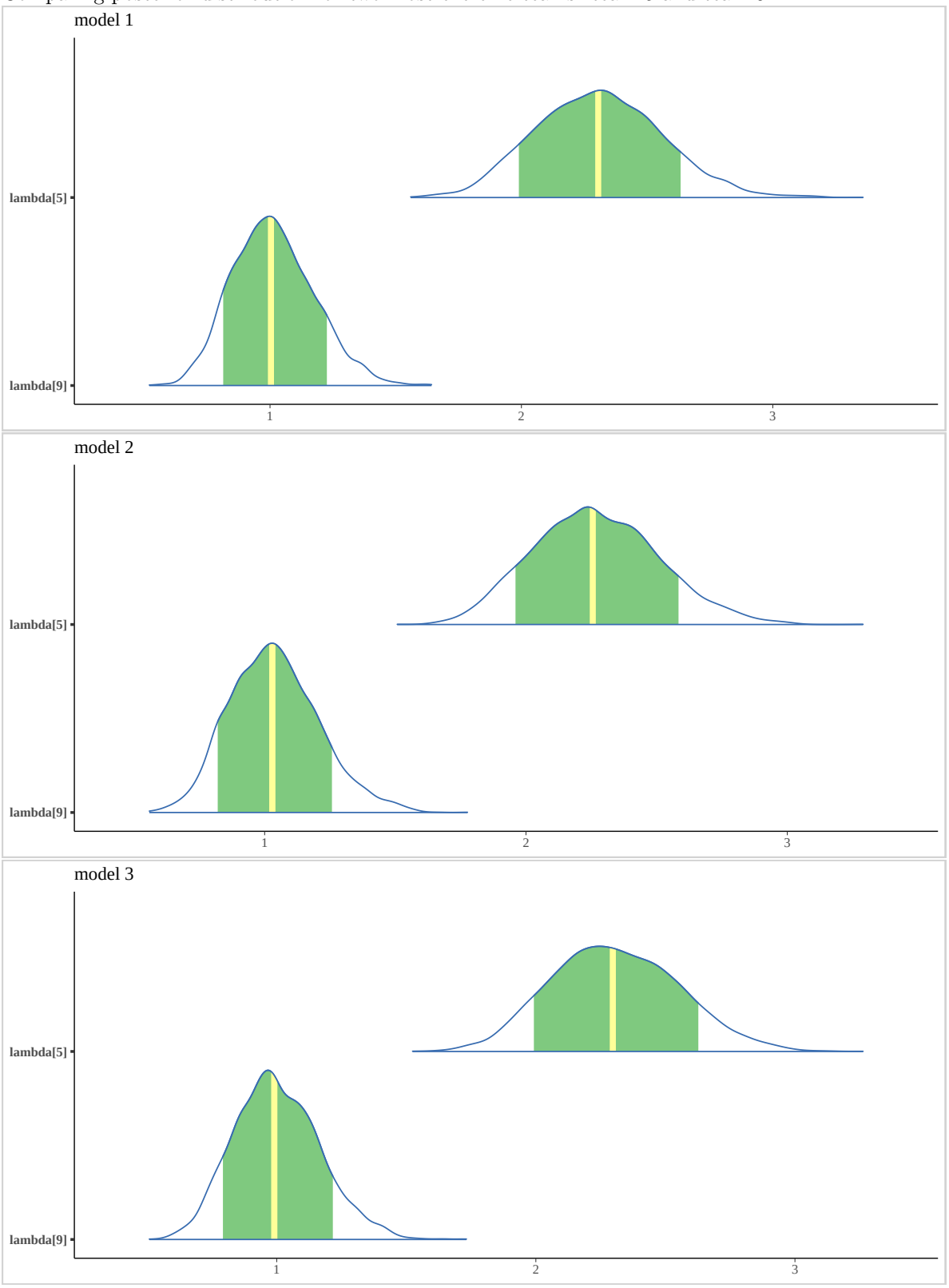


Posterior predictive checks

- All 3 models show similar posterior distributions. Means are roughly in the range 1-2.5.



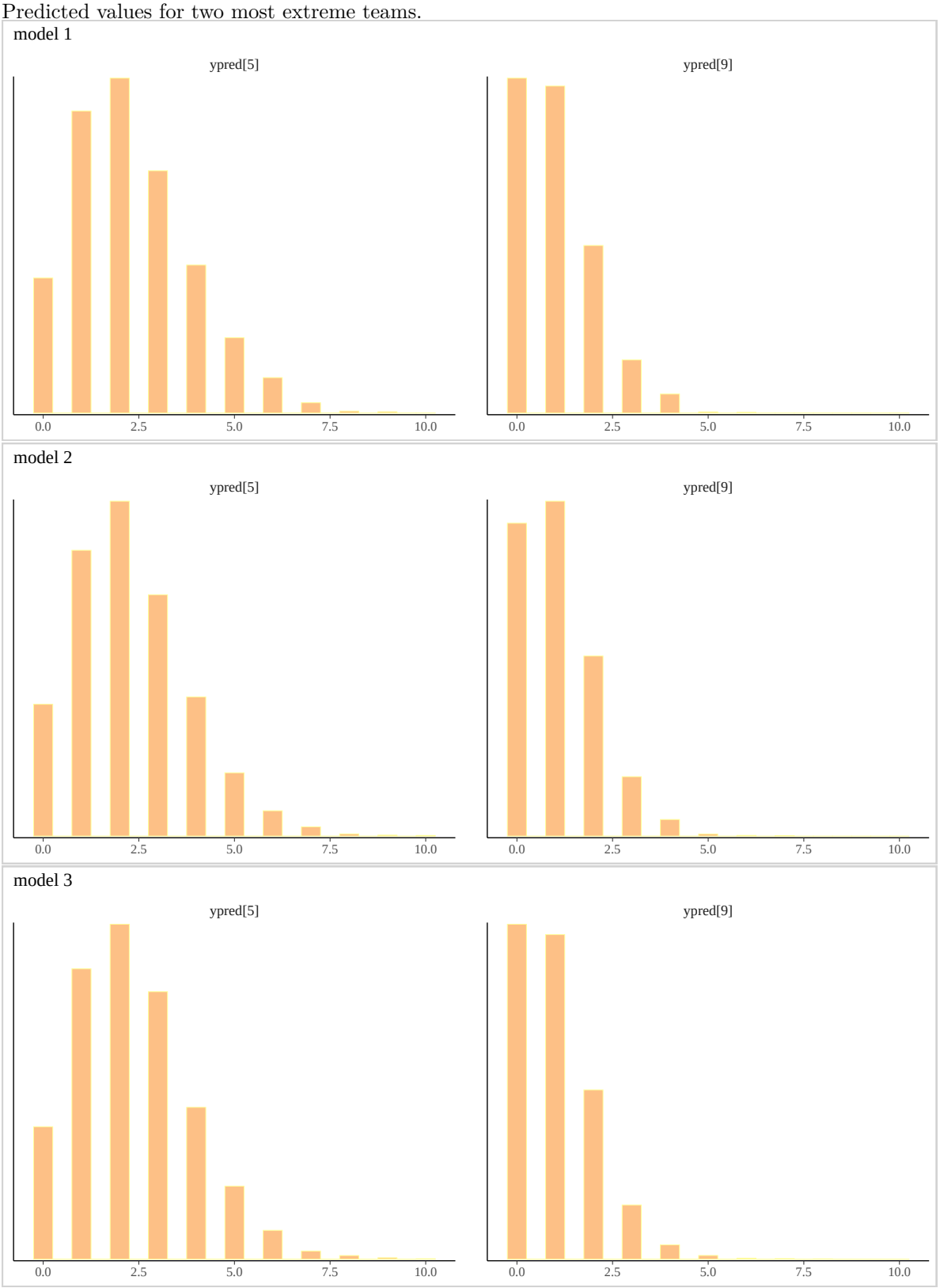
- Comparing posterior distribution for two most extreme teams: team 5 and team 9.



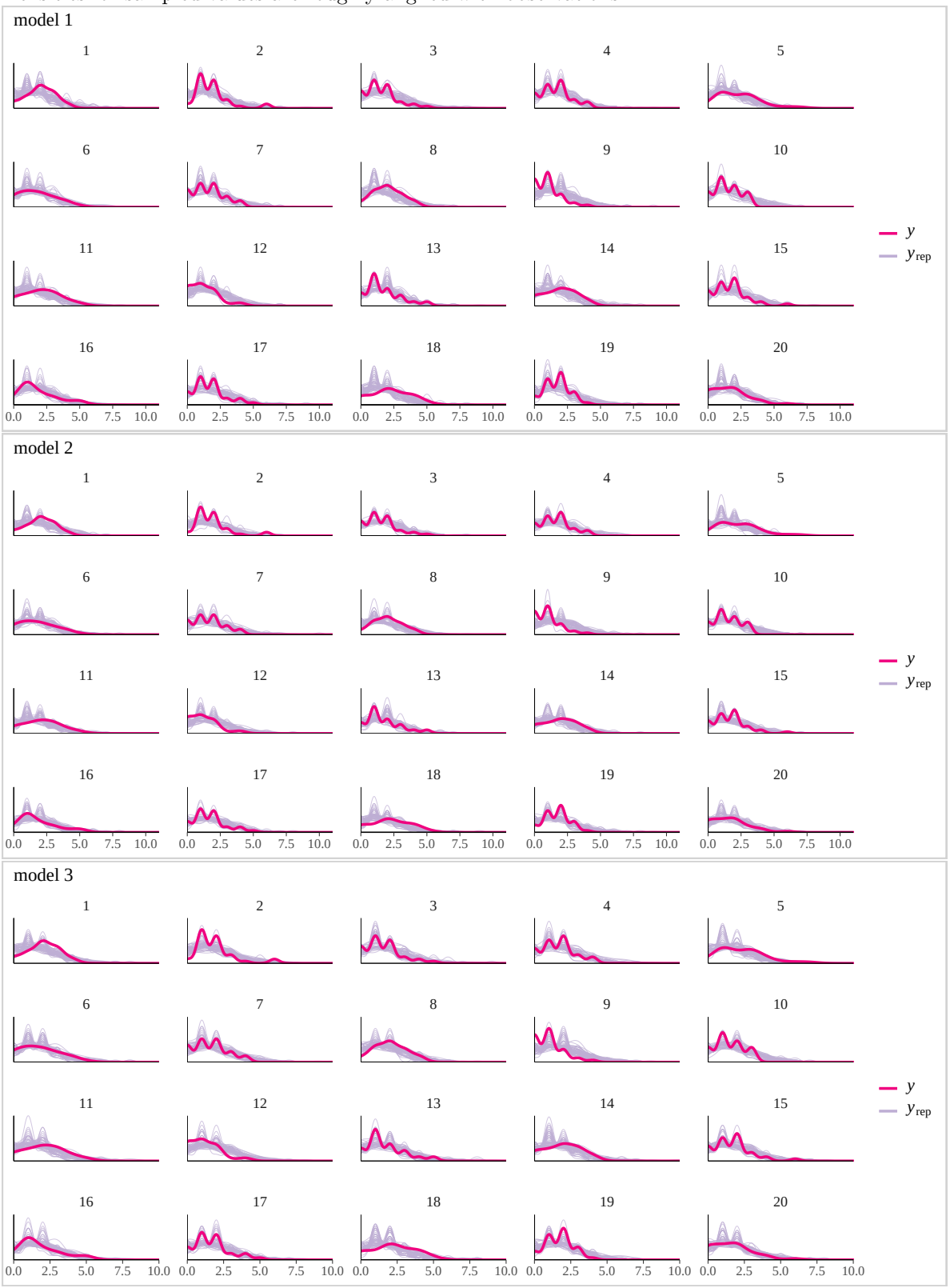
Predictive performance assessment

- Histograms for predicted values are similar to observed distribution.

- Predicted values for two most extreme teams.



- Densities for sampled values are roughly aligned with observations.



Model comparison

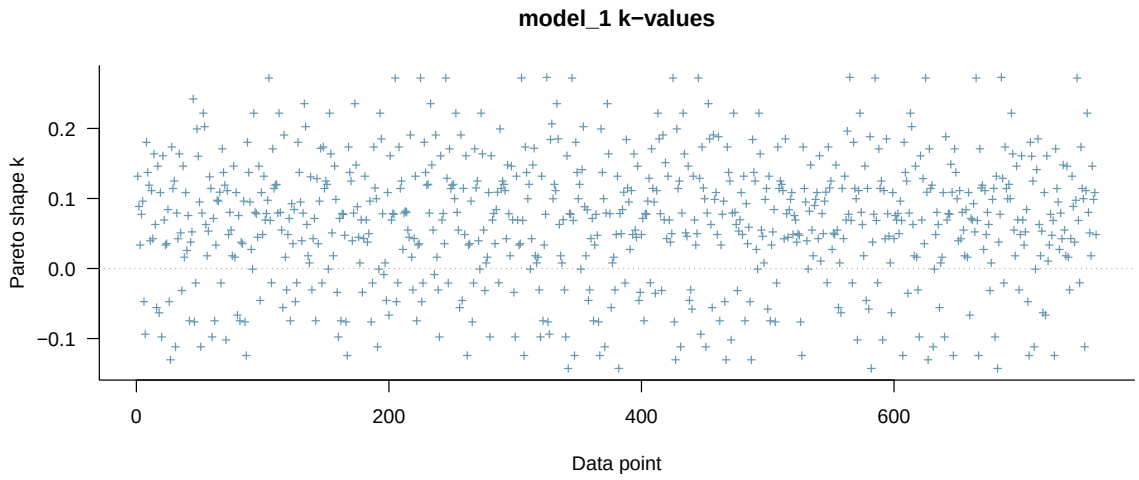
- Using LOO-CV, we can see that the models have very similar elpd_loo, meaning that they have a similar predicitive power. Model 2 (hierarchical) is just slightly better than the others.

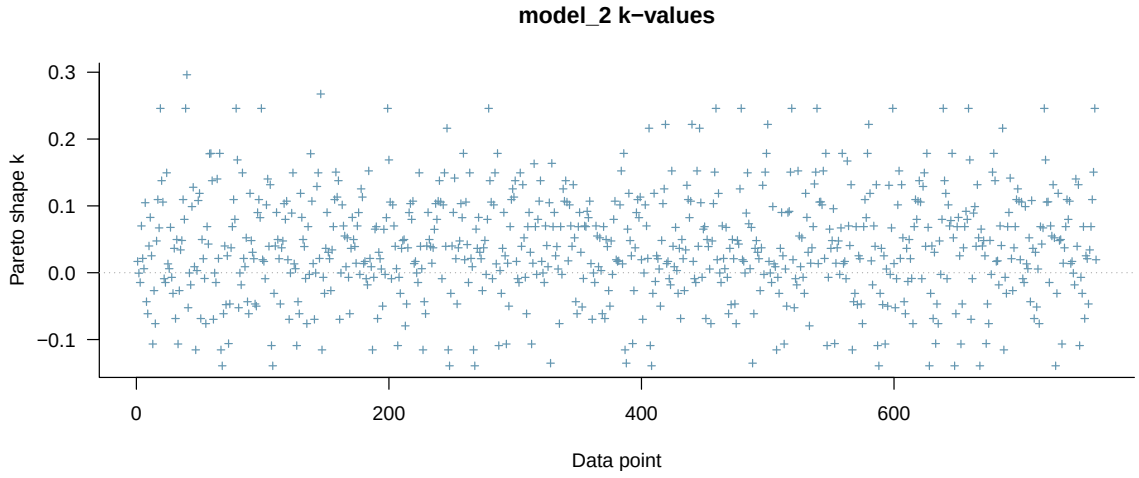
```
##
## Computed from 4000 by 760 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -1218.7 16.9
## p_loo        18.2  1.0
## looic      2437.4 33.9
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

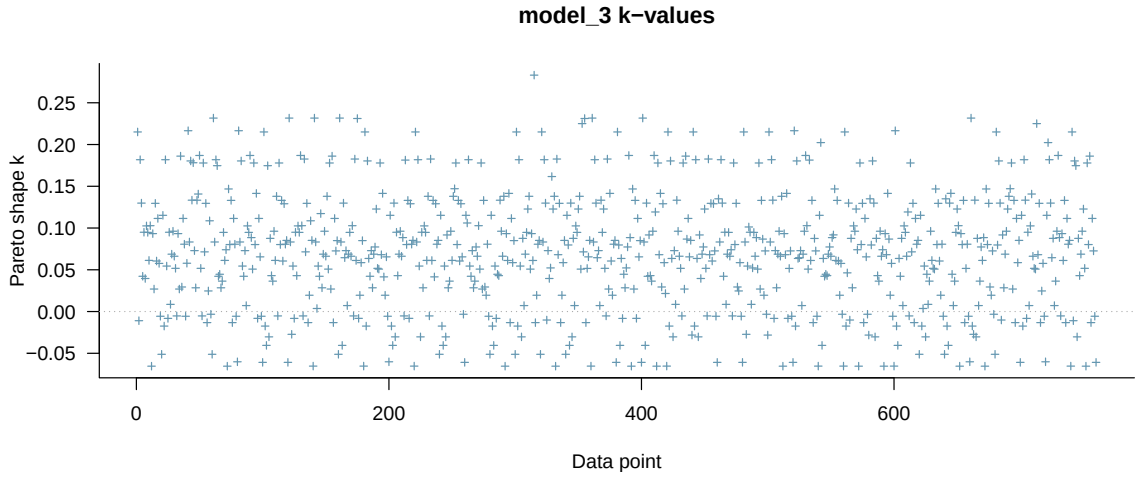
```
##
## Computed from 4000 by 760 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -1217.3 16.9
## p_loo        16.8  0.9
## looic      2434.6 33.7
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
##
## Computed from 4000 by 760 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -1219.2 17.1
## p_loo        18.7  1.0
## looic      2438.3 34.2
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

- For all models, k-values are good (k<0.5), meaning that our models have very reliable estimates.





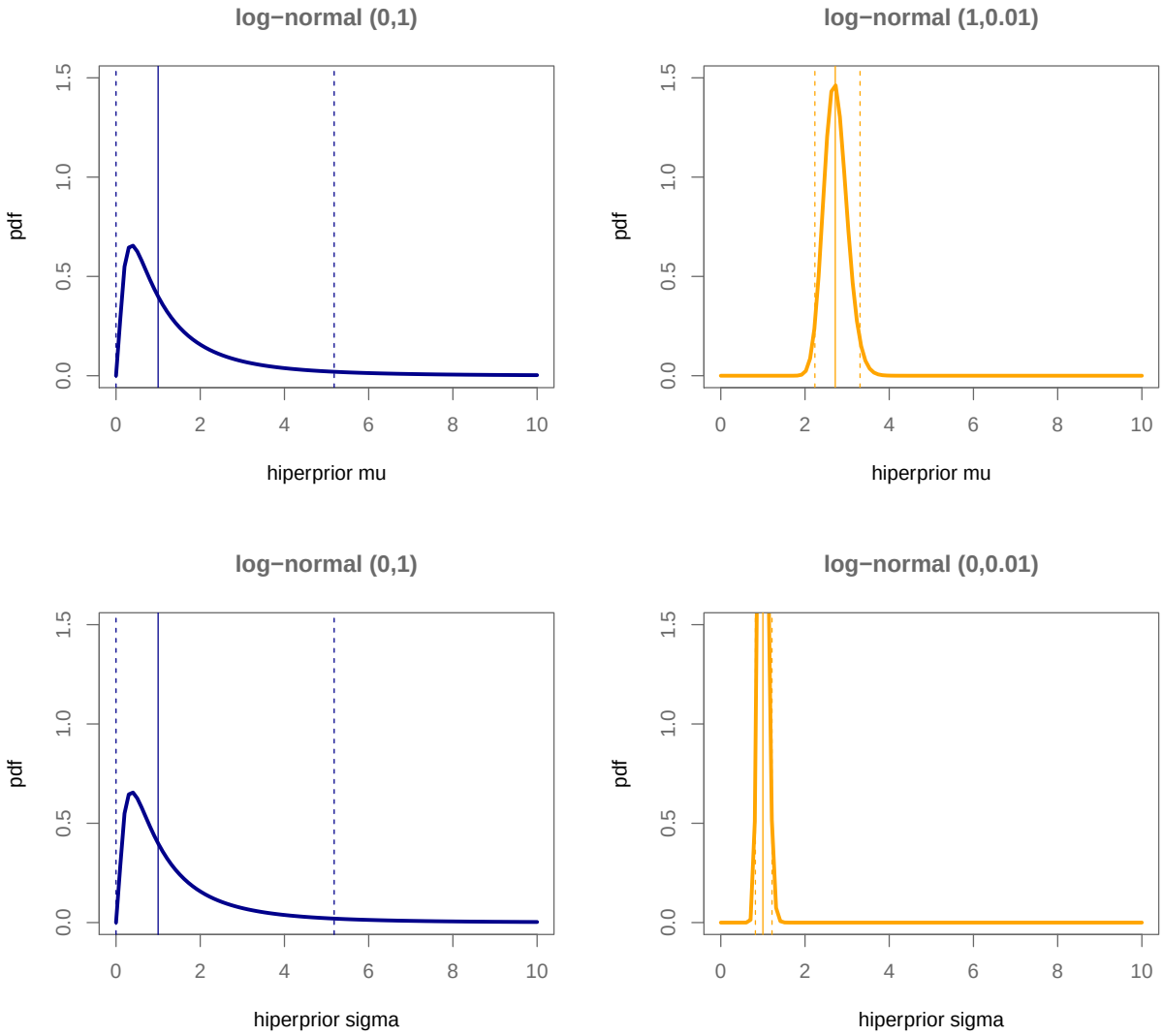


Sensitivity analysis with respect to prior choices

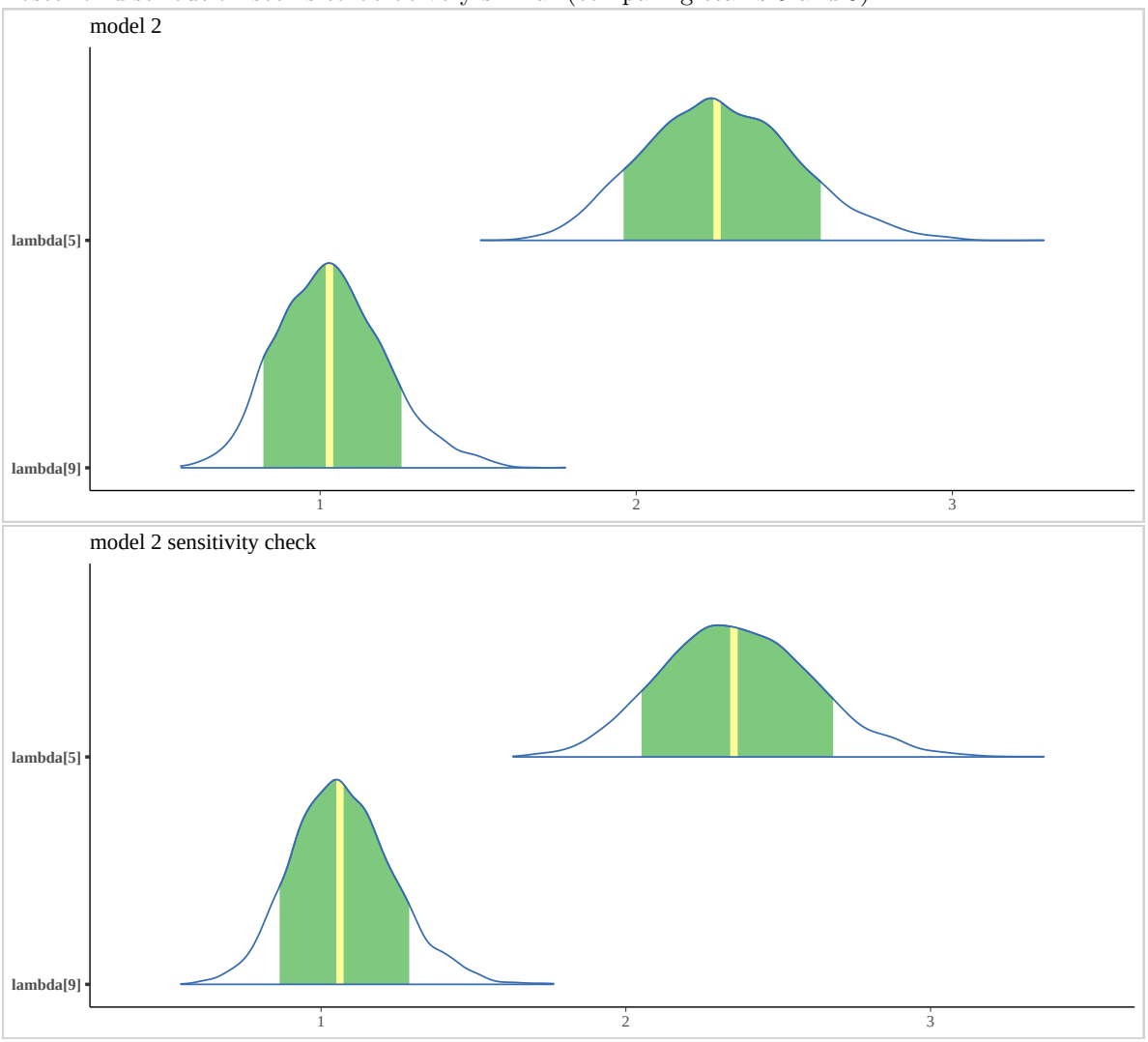
For sensitivity analysis, we are again running model_2, with much more informative priors:

$$\mu \sim lognormal(1, 0.01) \text{ instead of } \mu \sim lognormal(0, 1)$$

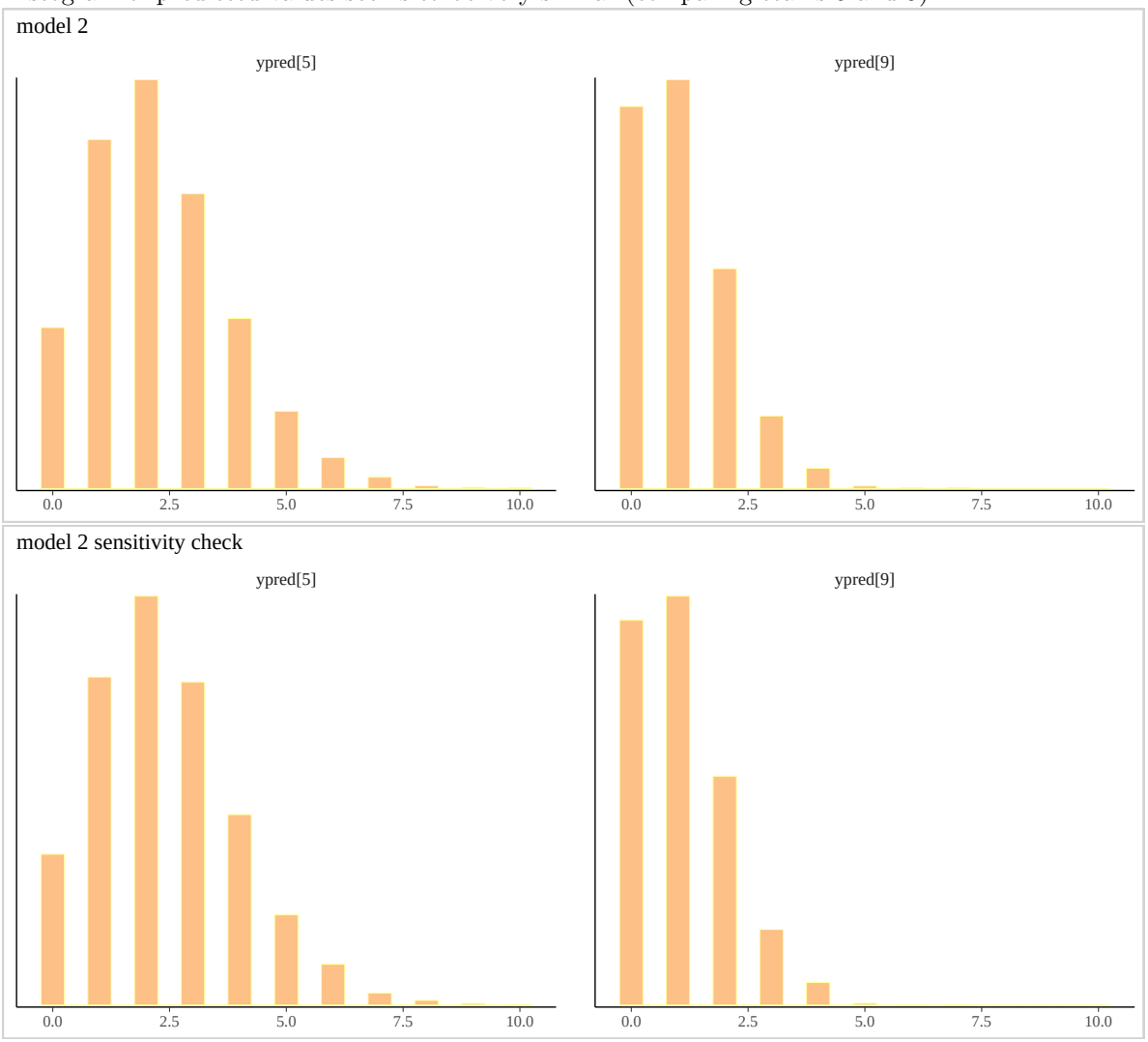
$$\sigma_i \sim lognormal(0, 0.01) \text{ instead of } \sigma_i \sim lognormal(0, 1)$$



- Posterior distribution seems to be be very similar (comparing teams 5 and 9).



- Histogram of predicted values seems to be very similar (comparing teams 5 and 9).



- Elpd_loo is slightly lower with the modified priors.

```
##
## Computed from 4000 by 760 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -1217.3 16.9
## p_loo        16.8  0.9
## looic      2434.6 33.7
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
##
## Computed from 4000 by 760 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -1218.8 16.2
## p_loo        17.7  0.9
## looic      2437.7 32.4
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

We can conlcude that model 2 is not very sensitive to the selection of priors.

Discussion

Issues

During the final project, we have faced some issues regarding the design of the models and practical implementation issues during modeling.

When we designed the model in the very beginning, we were uncertain about which prior distribution to choose and there have been a lot of discussions on this. After several rounds of testing different combinations separately and self-studying on how to build reasonable models, we finally reached persuasive solutions which are demonstrated in the modeling section. The teaching assistant was also crucial to help us out by guiding us to think about certain details.

In addition, there are also some practical challenges when implementing the models using Stan. For example, the issues regarding the input data type for certain sampling functions. We have spent days on some of these practical issues, and in the meanwhile also learned a lot when debugging.

It was a bit surprising to see that including a predictor did not improve predictive power. We assume that the reason lies in our choice of predictor value, and that there is no significant correlation between predictor and outcome. The team itself hierarchical group) is still the best predictor, and it has more influence on the outcome than what the other team did during the match.

Potential improvements

Three models are illustrated in the modeling section and they all achieved good performance. We also included one predictor of the opponent's yellow cards number in the 3rd model. Even though no obvious difference was observed, it was still a meaningful trial to take more environmental variables into account since football match is an interactive process within different teams.

Inspired by the Bayesian hierarchical model used for scoring prediction in Baio and Biangiardo's work[1], where the home advantage and the opponent condition are considered, we see a potential improvement of combining these factors also into the modeling of yellow cards prediction. An initial test model was implemented here as follows:

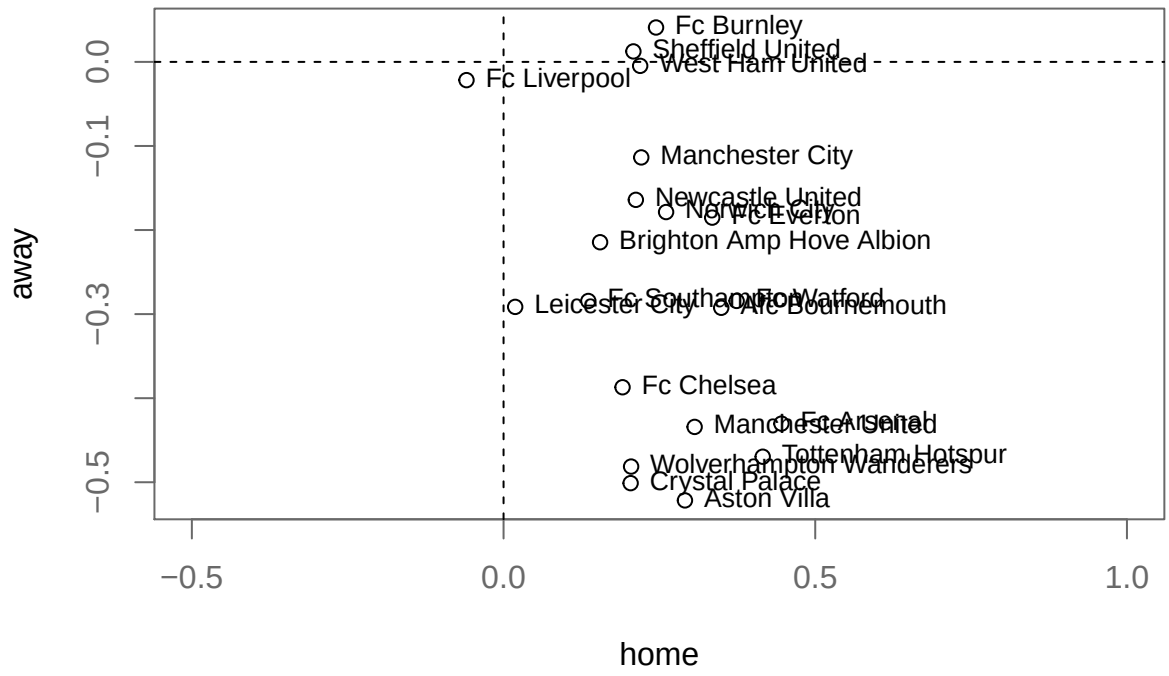
```
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
data_2 <- read.csv('datasets/yellow_card_home_away.csv')

ng = nrow(data_2) # number of games
nt = length(unique(data_2$home_team)) # number of teams
teams = unique(data_2$home_team) # convert teams name into a list of index
ht = unlist(sapply(1:ng, function(g) which(teams == data_2$home_team[g]))) # home team index
at = unlist(sapply(1:ng, function(g) which(teams == data_2$away_team[g]))) # away team index

stan_data_2 = list(
  nt = nt,
  ng = ng,
  ht = ht,
  at = at,
  y1 = data_2$home_yellow_cards,
  y2 = data_2$away_yellow_cards
)
model_4 = stan(file = 'models/potential_improvement.stan', data = stan_data_2,iter = 3000)
```

The following plot shows the tendency to get yellow cards considering the home/away situation of each team. As shown in the plot, FC Liverpool shows a relatively small value (close to zero) in both cases (home and

away). One possible explanation for the Liverpool team might be they play with a conservative strategy and prioritize safety. Further analysis could be conducted to validate the differences between teams but with this early test, we see potential improvement to combine more environmental factors into account.



- Stan code for **potential model including home/away and opponent team factors**:

```
## // The input data is a vector 'y' of length 'N'.
## data {
##   int<lower=0> nt; //number of teams
##   int<lower=0> ng; //number of games
##   int<lower=0> ht[ng]; //home team index
##   int<lower=0> at[ng]; //away team index
##   int<lower=0> y1[ng]; //yellow card home team
##   int<lower=0> y2[ng]; //yellow card away team
## }
## parameters {
##   vector[nt] home; //parameter for getting yellow cards as home team
##   vector[nt] away; //parameter for getting yellow cards as away team
## //hyper parameters
##   real mu_home;
##   real<lower=0> tau_home;
##   real mu_away;
##   real<lower=0> tau_away;
## }
## transformed parameters {
##   vector[ng] theta1; //probability of getting yellow cards as home team
```

```
##   vector[ng] theta2; //probability of getting yellow cards as away team
##   theta1 = exp(home[ht] - away[at]);
##   theta2 = exp(home[at] - away[ht]);
## }
## // The model to be estimated. We model the output
## // 'y' to be normally distributed with mean 'mu'
## // and standard deviation 'sigma'.
## model {
## //hyper priors
## mu_home ~ normal(0,0.1);
## tau_home ~ normal(0,1);
## mu_away ~ normal(0,0.1);
## tau_away ~ normal(0,1);
## //priors
## home ~ normal(mu_home, tau_home);
## away ~ normal(mu_away, tau_away);
## //likelihood
##     y1 ~ poisson(theta1);
##     y2 ~ poisson(theta2);
## }
## generated quantities {
## //generate predictions
##   int<lower=0> y_home[ng]; //yellow card home team
##   int<lower=0> y_away[ng]; //yellow card away team
##   real log_like_home[ng];
##   real log_like_away[ng];
##   for (i in 1:ng){
##     y_home[i] = poisson_rng(theta1[i]);
##     y_away[i] = poisson_rng(theta2[i]);
##   }
##   for (i in 1:ng){
##     log_like_home[i] = poisson_lpmf(y1 | theta1[i]);
##     log_like_away[i] = poisson_lpmf(y2 | theta2[i]);
##   }
## }
```

Conclusion

Three models implemented in this final project all achieved good performance. The results are validated through posterior checks and predictions are compared to real results and expected distribution of yellow cards rate for certain teams. The research question of predicting yellow cards is answered properly and future improvements are suggested.

We see the potential and meaning of predicting yellow cards, which helps decision-making and strategy adjustment for different teams in football matches.

Based on our modeling and observations, here we propose two hypotheses that could be tested in the future. First, the yellow cards rate for each team might be related to aggressiveness. Second, the lower probability of getting yellow cards might be explained by a conservative and safe strategy of playing. This strategy could be related to the price of the football players - the more expensive they are, the more careful they will be in playing football matches.

Self-reflection

During this group project, we practiced what we learned from the course with real scenarios, where a Bayesian model can be made to make predictions and assist decision-making. Driven by the interest in football matches, we started the project by searching for available datasets and then developed the initial understanding of the dataset and designed the model together through a series of remote meetings.

While conceptualizing the model, we did meet some problems regarding how to build a meaningful model and what priors and distributions should be chosen as proper approaches. Unlike the assignments, where priors and assumptions are provided, all the details need to be considered thoroughly and justified. To validate our idea, we found certain evidence either based on our football pre-knowledge or observations from the dataset, and discussed with each other to reach a consensus. Through the iterations, we developed a good understanding of the dataset and the complete process of applying Bayesian data analysis.

This experience benefits us a lot not only for future work when dealing with similar projects but also for a way of thinking to decompose problems and develop reasonable solutions. Overall, we think we have achieved satisfying results and it was really an enjoyable experience to cooperate.

Reference

[1]Baio, G., & Blangiardo, M. (2010). Bayesian hierarchical model for the prediction of football results. Journal of Applied Statistics, 37(2), 253-264.