Predicting Allsvenskan – Bayesian modeling of football results

Gustav Sternelöv

May, 26th, 2016

Contents

1 Introduction

1.1 Project description

A very popular part of the sports industry is to predict future outcomes. In football the predictions often concerns which team that will win next game, how many goals will each team score and which team will have the most points at the end of the season. To be able to predict outcomes of this sort an estimate of how good the respective teams are in relation to each other is needed since the teams in a football league not are equally skillful. Instead, the difference in team strength is the factor with the most influence on the results.

The aim with this report is to measure the skill, or strength, of the teams in Allsvenskan, Sweden´s top football division. These measures of team skill will then be used for making predictions on the outcomes in future games. To model the strength of the teams in Allsvenskan Bayesian techniques will be used. More specifically, a so-called hierarchical Bayesian model will be estimated for measuring the team strength.

Similar types of studies with hierarchical Bayesian models has been conducted for other football leagues and these studies are highly inspirational for this study. A few examples of studies where this kind of model has been used for measuring team strength and making predictions is presented in the next chapter.

1.2 Earlier studies

Blangiardo and Baio [2010] used match results for the 1991-92 season in the Italian Serie A to build a hierarchical Bayesian model with the aim of producing predictions of football results. The authors measured both the attacking and defensive skill of each team and used these estimates for predicting match results. Apart from the skill parameters they also estimated a parameter *home* which modeled the advantage of playing at home. The authors found that the effect of playing at home was positive. Moreover, the league winners AC Milan had the highest estimate for the attacking parameter and was also one of the clubs with the best defensive parameter estimate. Among the teams with weakest defensive capability were the relegated side Ascoli. The predictive ability of the model were tested by simulating the results of the 1991-92 season and comparing with the actual results. In general, the authors concluded, the hierarchical Bayesian model did seem to be a rather good fit.

At the blog *Pass the ROC* (Weitzenfeld [2014]) used the model constructed by Blangiardo and Baio as a starting point for modeling football results with 2013-14 season of the English Premier League as data set. The author modified the model of Blangiardo and Baio a little bit by tweaking the priors and including an intercept. Weitzenfeld also concluded that there is a significant advantage of playing at home. He also found it to be a good idea to include an intercept since the HDP interval (Highest posterior density interval) for the parameter not included zero. Just as Blangiardo and Baio, Weitzenfelds estimates of the average attacking/defensive were clearly correlated with the teams' positions in the league. Weitzenfeld checked the predictive ability of the model by simulating the 2013-14 season and comparing the actual number of goals during the season with the simulated number of goals. He noted that the model worked rather well, but that shrinkage toward zero for the attack parameter had some implications on the estimates. For the high-scoring teams the model therefore overshrinked the estimates and predicted these teams to score slightly fewer goals than they actually did. Reversely, for the low-scoring teams, the model estimated them to score more goals than they actually did.

A third example of where a hierarchical Bayesian model has been used for predicting the outcomes of football games is in the paper *Modeling Match Results in La Liga Using a Hierarchical Bayesian Poisson Model*(Bååth [2013]). Bååth's model is a bit different to the two earlier ones in the sense that he measures the team strength by a single skill parameter instead of with both an attack and a defense parameter. To model the home-advantage he specified two different intercepts, one for the team playing at home and one for the away team. Bååth also found that there is a significant home advantage and that it corresponds to, on average, almost 0.5 more goals for the home team. The author then compares the expected number of goals when playing at home for all teams given the skill parameter to examine how well the teams are ranked by the parameter. This ranking agrees well with actual ranking and relation between the teams in the league. For

testing the predictive ability of his model Bååth then examine how well the simulations agrees with the actual number of goals in game and the actual results. He finds that his model on 34 % of the times predicts the correct number of home goals and that it predicts the right match outcome 56 % of the time. He then also simulates results for all remaining games of the 2012-13 season of La Liga but does not compare it with the actual outcomes.

1.3 Data

In the data set are all the match results for the 2015 season of Allsvenskan and the results for the first twelve rounds of the 2016 season. For the remaining games of the 2016 are there no results since these games not have been played yet. There are 16 teams in Allsvenskan and they play each other twice in a season, on time at home and one away. After the 2015 season two teams were relegated (Åtvidabergs FF and Halmstads BK) and replaced by two teams (Östersunds FK and Jönköpings Södra) from the second highest division, Superettan. As a result of that, the data set includes match results for 18 different teams and the number of observations vary between the teams. For the relegated sides there are 30 observations and for the new teams 12 observed results, the other teams have 42 observed match results each. The data set contains four columns which are *HomeTeam*, *AwayTeam*, *HomeGoals* and *AwayGoals* and the first six observations are shown below.

```
##        Home.Team       Away.Team Home.Goals Away.Goals
## 1    Hammarby IF       BK Häcken          2          0
## 2      Kalmar FF Helsingborgs IF          0          0
## 3 Falkenbergs FF        Gefle IF          0          2
## 4   IFK Göteborg  Åtvidabergs FF          1          0
## 5 Djurgårdens IF     IF Elfsborg          1          2
## 6 IFK Norrköping       Örebro SK          1          1
```

2 Method

2.1 The model

The number of goals scored in a game or by each team can be shown to be Poisson distributed. This is proven in a lot of articles, for instance by Baio and Blangiardo[2010] in the article mentioned earlier in the report.

$$Goals \sim Poisson(\lambda)$$

Thus, the expected number of goals for the home and away team are then modeled as being Poisson distributed in the following way.

$$y_{gj}|\theta_{gj} \sim Possion(\theta_{gj}$$

The number of goals scored by the home team is denoted as y_{gh} and for the away teams as y_{gh}. The notation g represents the gth game in the season, or as in this example the gth game in the sample of games since the data set contains more than a full season.

The θ parameters represents the average number of goals scored in a game by the home team and the away team. For modeling these parameters is a log-linear random effct model proven to be a suitable choice (for example, Baio and Blangiardo[2010]). Then, the expressions for the respective log θ parameters are specified as follows.

$$log\theta_{gH} = Intercept_{Home} + Skill_{Home} - Skill_{Away}$$
$$log\theta_{gA} = Intercept_{Away} + Skill_{Away} - Skill_{Home}$$

For the log θ:s are two different intercepts estimated, one for the home team and one for the away team. In that way is the advantage of playing at home included in the model. The intercepts shall be interpreted as

the log average number of goals in a game when both teams are equally good.
A self-evident assumption to make is that the teams in Allsvenskan are on different levels where some teams are more skillful and some teams are less skillful. However, this variable of actual skill of the teams is a latent variable since it not is possible to observe it directly. Instead earlier match results are collected for estimating the skill parameter for each team. The prediction for the number of goals a team will score is then given by the intercept plus the skill of the team minus the skill of the opponents.
The priors for the intercepts are set to be very vague in all examples mentioned in the *earlier studies* chapter. Perhaps they not have to be that vague, but on the other side do I not feel confident enough to set a prior which not would be considered as vague.

$$Intercept_{Home} \sim Normal(0, 4^2)$$

$$Intercept_{Away} \sim Normal(0, 4^2)$$

The prior for the skill parameter is normally distributed with the hyper-priors μ_{Teams} and σ^2_{Teams} as parameters.

$$Skill_{1,\ldots,T} \sim Normal(\mu_{Teams}, \sigma^2_{Teams})$$

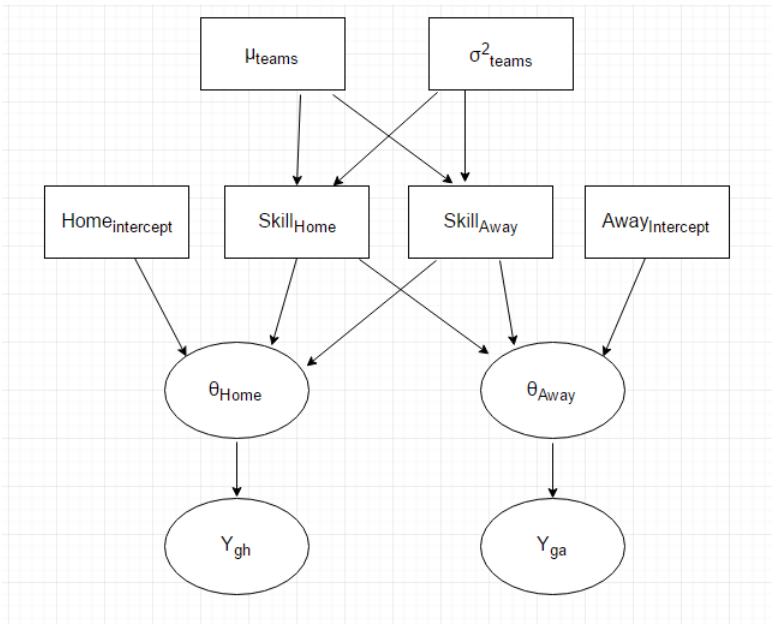
The hyper-priors for the skill parameter are also chosen to be vague:

$$\mu_{Teams} \sim Normal(0, 4^2)$$

$$\sigma_{Teams} \sim Normal(0, 4^2)$$

The decision to set the mean for μ_{Teams} to zero may cause a bit of overshrinkness for the skill estimates. However, as mentioned earlier, this was mainly a problem for the teams at the very top of the Premier League who were a lot better than the other teams during that season. Allsvenskan on the other hand is well-known for being a very even league with a lot of teams competing for the title instead for just a few. If this fact lowers the risk of overshrinkage is hard to tell, but it might be interesting to compare and investigate if it seem to have any effect.

A clearer view of the model can be given by a DAG representation of the models hierarchy:



The two hyper-parameters in the model, μ_{Teams} and σ_{Teams}, builds a latent structure and this structure is assumed to be representive for all the games played in the data set. The use of the hyper-parameters is to obtain an estimate of the average scoring rate. In the estimation of the parameters in the model are all games contributing. This means that the parameter estimate of skill for a team with few observations uses the hyper-parameters μ_{Teams} and σ_{Teams}, which uses all data. The effect of this is that even for a team that not has played that many games it is possible to get rather reasonable estimates.

The samples of the parameters are obtained by Gibbs sampling. To constrain how the estimates of team skill evolves during the iterative process is one of the skill parameters fixed at a constant value (zero). The effect of this is that the estimates of skill for the other teams will be interpreted as relative to the team with the skill estimate held fixed at zero.

2.2 Expected number of goals

An interesting result might be the estimate of how many goals a team is expected to score. This can be computed by using the estimations for the skill parameters but first they need to be modified in order to become easier to interpret. Since one of the skill parameters are fixed at zero all of the other parameters are relative to the skill of that team. To make the interpretation easier the skill parameters are centered as the mean skill of all teams are subtracted and either of the home or away intercept is added. If the home intercept is added the expected number of goals when playing at home is obtained and otherwise the expected number of goals when playing away from home.

2.3 Highest posterior density intervals (HPD intervals)

As described in the textbook *Bayesian data analysis* (Gelman et al. [2013]) is it in most cases important to present the posterior uncertainty together with point estimate for a parameter. To summary the uncertainty is often an interval calculated and one of these possible intervals is called the *Highest posterior density interval* (HPD interval). This interval contains the set of values which includes the 100(1-α) % of the posterior probability and will in this report be used for measuring the uncertainty of the parameter estimates.

3 Software

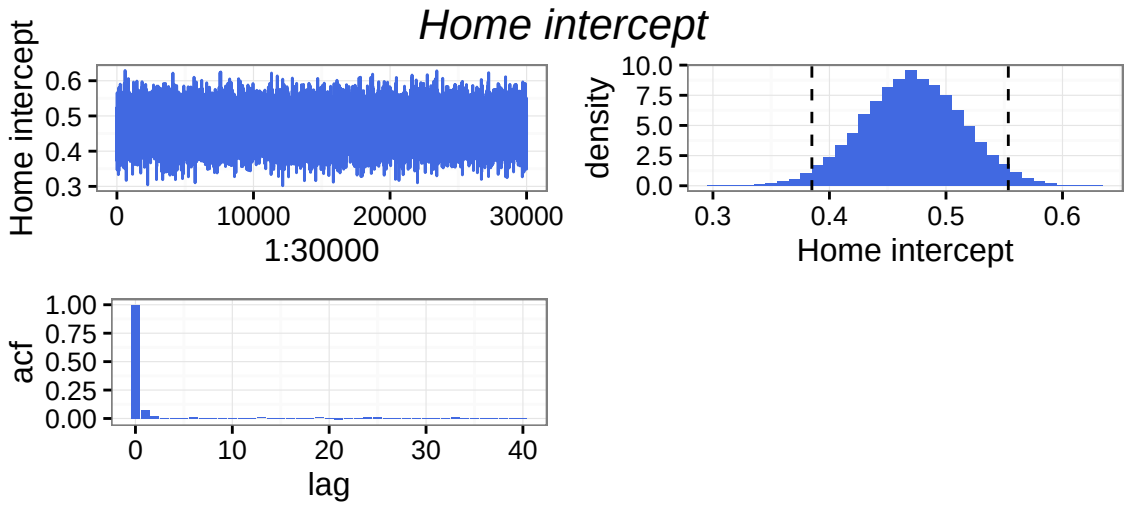
For running the Markov chain Monte Carlo method called Gibbs sampling have I used the packages *rjags*(Plummer [2016]) and *coda*(Plummer et al. [2006]). The visualizations of the results has been created with *ggplot2*(Wickham [2009]).

4 Results

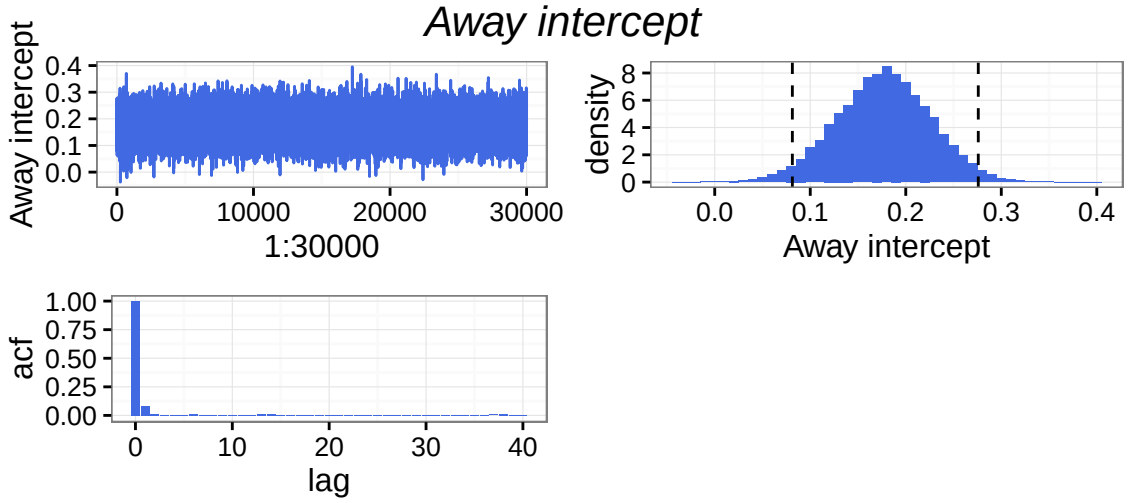
In the results section will first the Markov chains obtained by the Gibbs sampler be presented. Secondly, the skill in terms of expected goals er game for each team is presented and lastly will the predictive ability of the model be examined.

4.1 Analysis of convergence

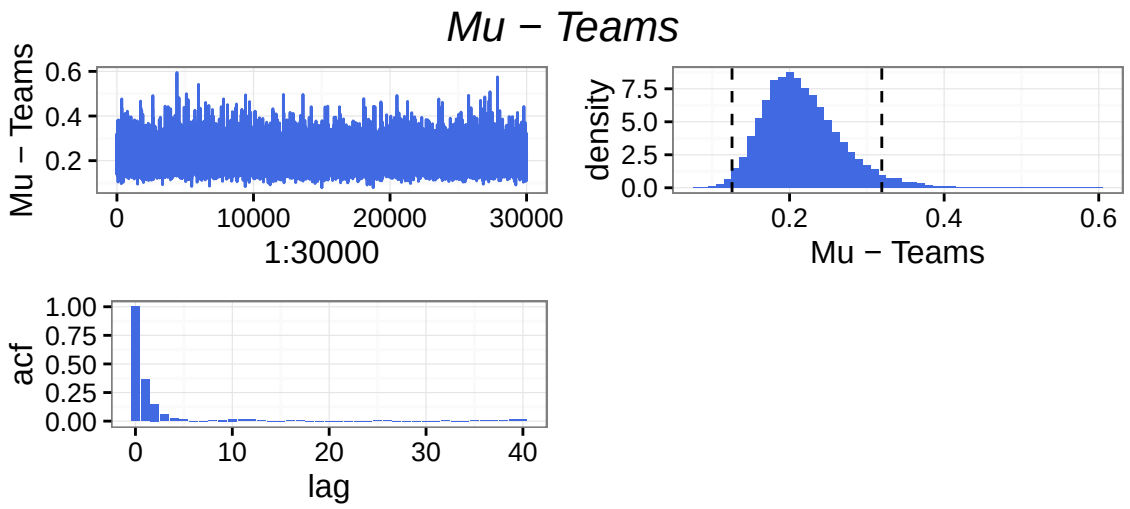
A Gibbs sampler is used for obtaining the chains of generated values for all the parameters in the model. The sampler generates in total 60 000 values and half of the values are kept after the burn-in period has been discarded. For examining if the chains has converged are trace plots, histograms and auto-correlation plots produced for each parameter. In the histograms are 95 % HPD intervals added to the plots.



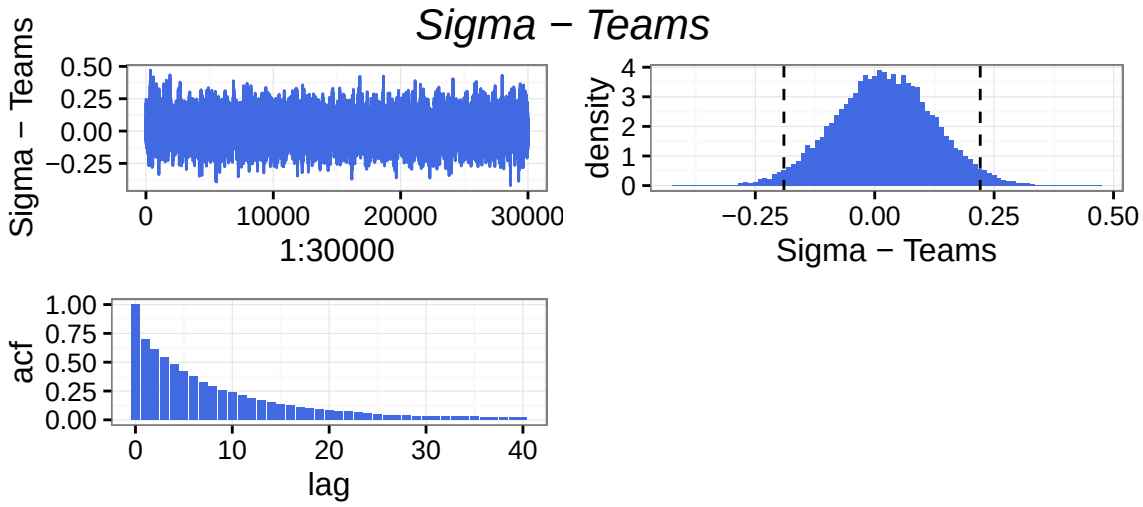
The interpretation of the trace plot for the chain of the parameter home intercept is that it has reached convergence. The pattern is random over all the chain and it does not get stuck for any visible periods. In the histogram it can be seen that the generated values are normally distributed with a 95 % HPD interval between 0.38 and 0.55. The correlation between the generated values in the chain is very low as can be seen in the plot with the auto-correlations for lag 0 to 40. Hence, the chain is interpreted to have converged and the low correlation speaks for the efficiency of the sampler.



The plots for the chain of the parameter for the away intercept are very similar to the plots for the home intercept. Neither for this chain is it then any doubts over the convergence or efficiency. The 95 % HPD interval for the away intercept is 0.08-0.28, meaning that the log average number of goals is higher for the team playing at home than for the team playing away.

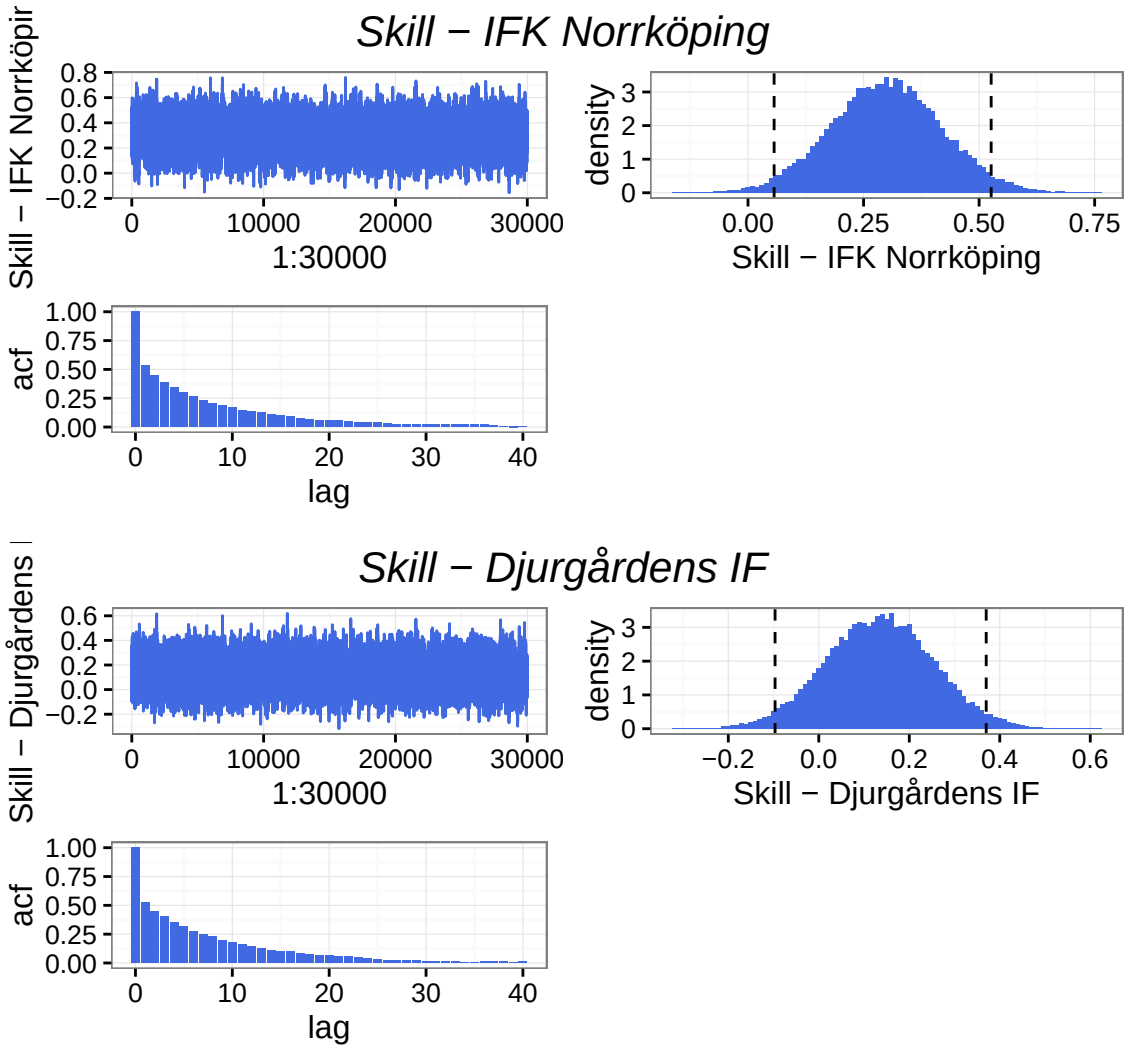


For the parameter μ_{Teams} the trace plot indicates that the chain has converged. The histogram strengthens this view of the convergence and regarding the efficiency it can be seen that the auto-correlation is very low.



No visible differences are seen for the trace plot and histogram for the σ_{Teams} parameter compared to the chains for the other parameters. The auto-correlation is a bit higher and, even though it dies out fairly quickly, the chain is less efficient in comparison to the other chains.

Since each team has its own skill parameter are 17 different team skill chains generated (one team skill parameter held fixed at zero). The plots for two of the teams are presented here below and the plots for the remaining teams can be seen in the appendix.



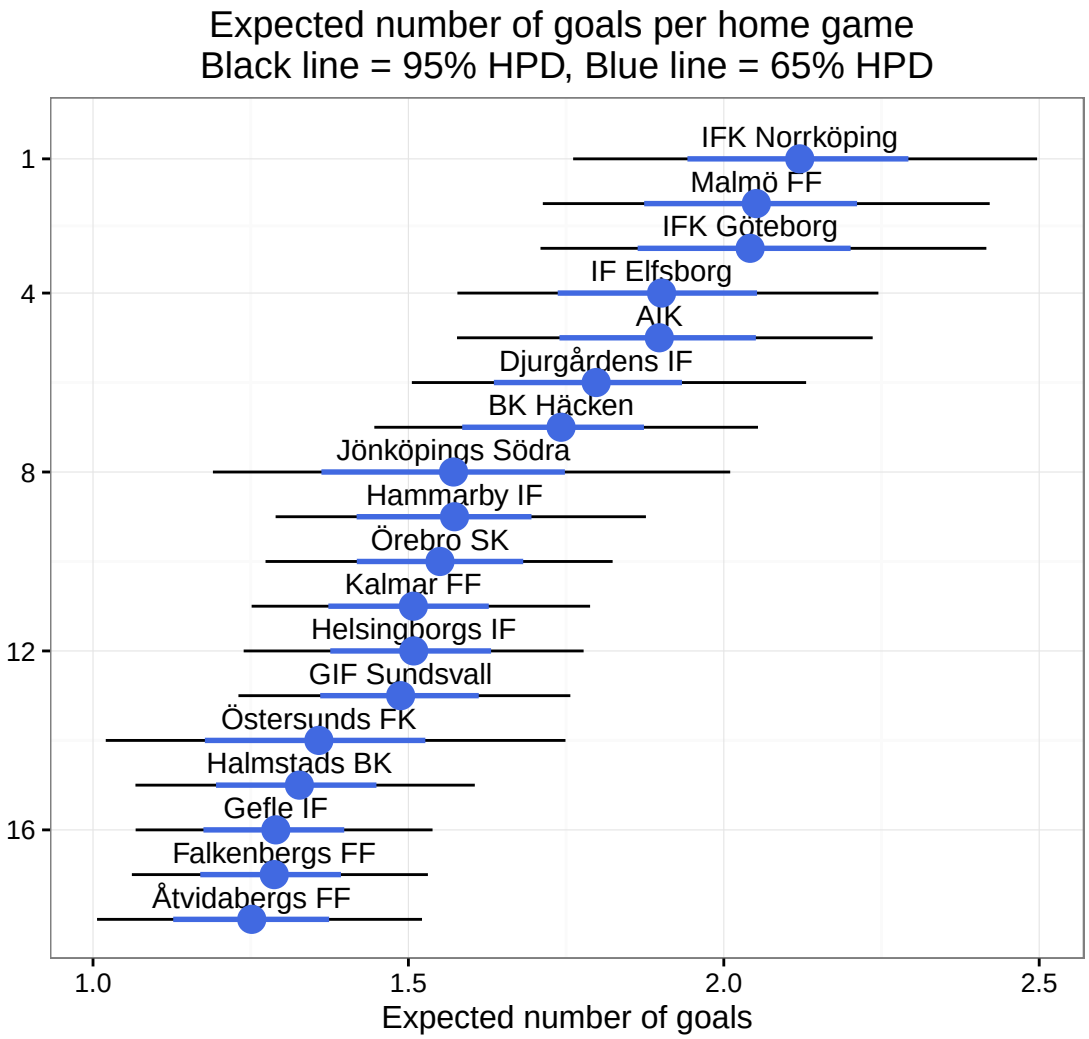
The trace plots are very similar, settled with a random pattern, for both the skill parameter for Norrköping and the one for Djurgården. That the chains has converged is also indicated by the histograms where the generated values seem to be normally distributed. The auto-correlation is for both chains quite high for the first lags but does then die out fairly quickly. The trace plots, histograms and auto-correlation plots are very similar for the chains of the skill parameter for the other teams so no further analysis is performed on these chains.

4.2 Measure of team skill

To examine how the model rank the teams in Allsvenskan is the team skill in terms of expected number of goals per goals calculated and presented. Since the model has two different intercepts, one for the team playing at home and one for the team playing away, the average number of goals a team is expected to score differs depending on if it is a home game or an away game.

4.2.1 Home

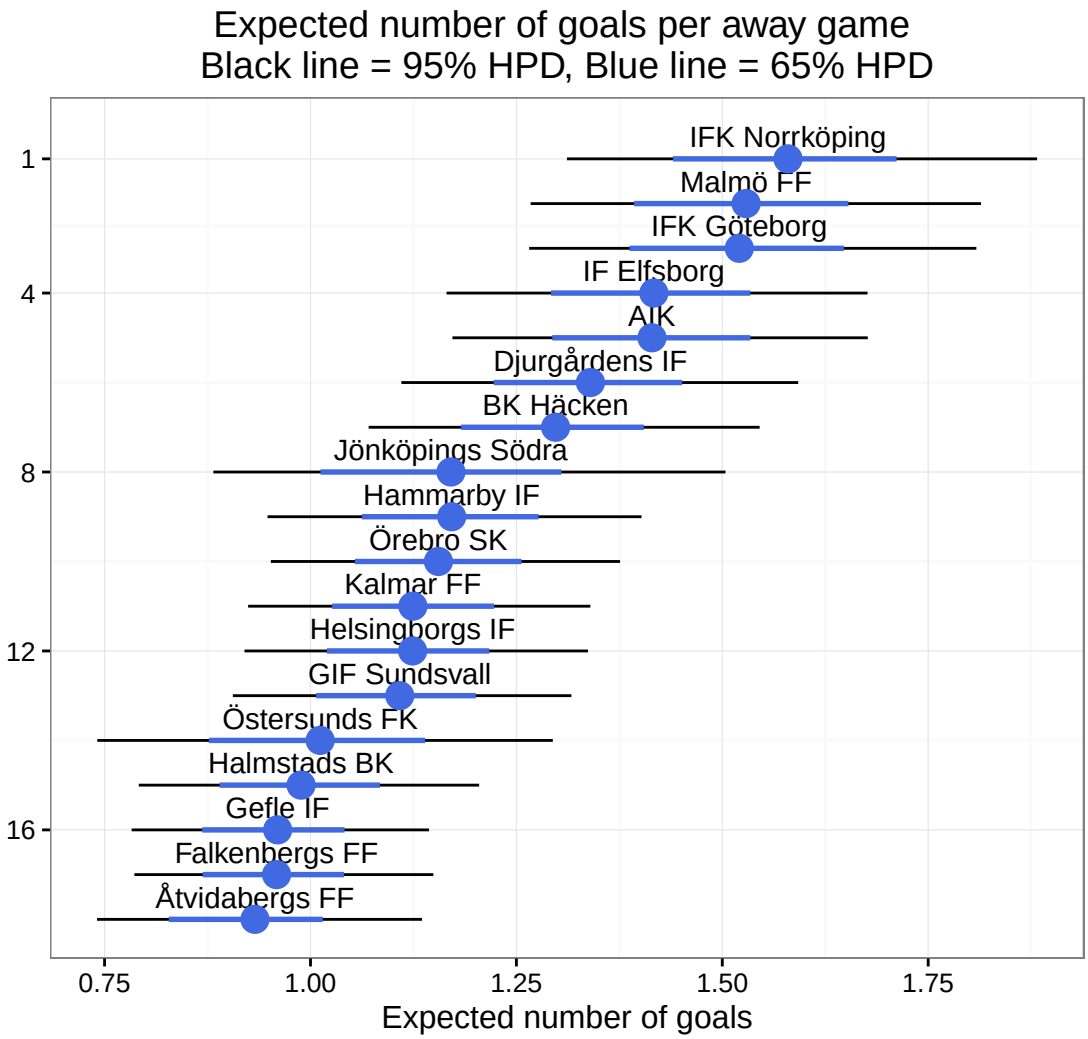
The expected number of goals when playing at home is visualized with the following plot. The blue dot is the median number of goals, the thick blue line is a 65 % HPD interval and the thin black line is the 95 % HPD interval.



The winners of the 2015 season, IFK Norrköping, are the team with the highest number of expected goals per game. Malmö FF, the current leader of the 2016 season and fifth best team last season, has the second highest median and after them comes IFK Göteborg, Elfsborg and AIK. These teams were all in the top 5 at the end of the 2015 season and none of the teams are placed lower than seventh place after the first twelve round of the 2016 season. The 95 % HPD intervals overlap for most of the teams and for IFK Norrköping you have to go down to the sixth placed team, Djurgården, before the 65 % HPD interval no longer overlaps. Found at the bottom of the ranking are the relegated teams of the 2015 season, Åtvidaberg and Halmstad, and the teams currently in the relegation zone, Gefle and Falkenberg. Noteable is also that the newcomers for the 2016 season, Jönköping and Östersund, have HPD intervals that are a bit wider than the intervals for the other teams.

4.2.2 Away

The expected number of goals when playing away from home is visualized with the same type of graph as the home goals.



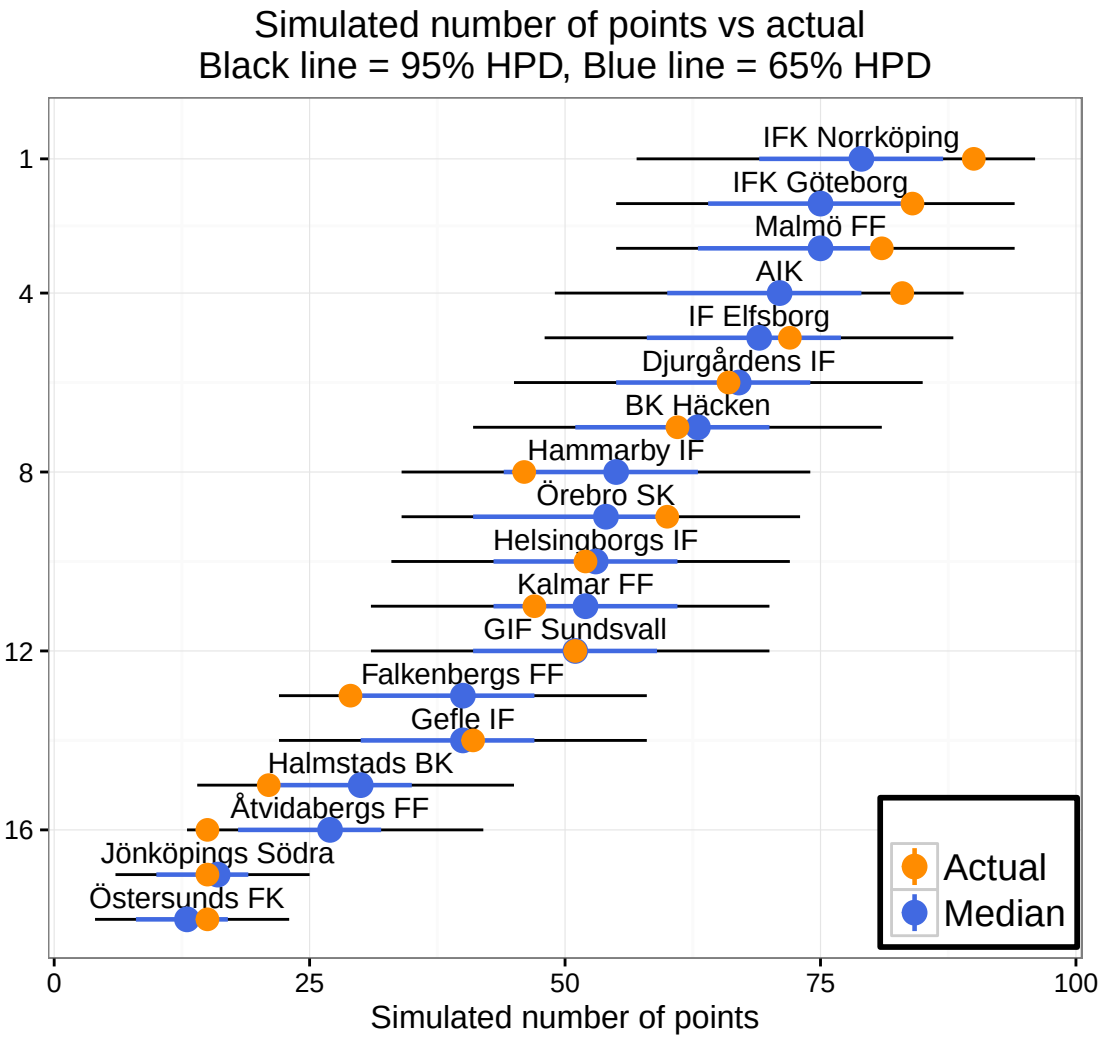
The expected number of goals is clearly lower when the teams are playing away from home. For instance, IFK Norrköping are expected to score 2.12 goals per game with the 95 % HPD interval 1.76-2.50 when playing at home and 1.58 goals per game with the 95 % interval 1.31-1.88 when playing away from home. However, apart from the difference in the number of goals is the ranking and the intervals very similar to when the teams are playing at home.

4.3 Predictions

The predictive ability of the model is evaluated by first running 30 000 simulations of the 336 games in the data set that already has been played. How many points and goals the teams are predicted to have gathered according to the model is then compared against the actual results. In the second step is the remaining 144 games of the 2016 season simulated and summarized in a table which gives the predicted final standings for the season.

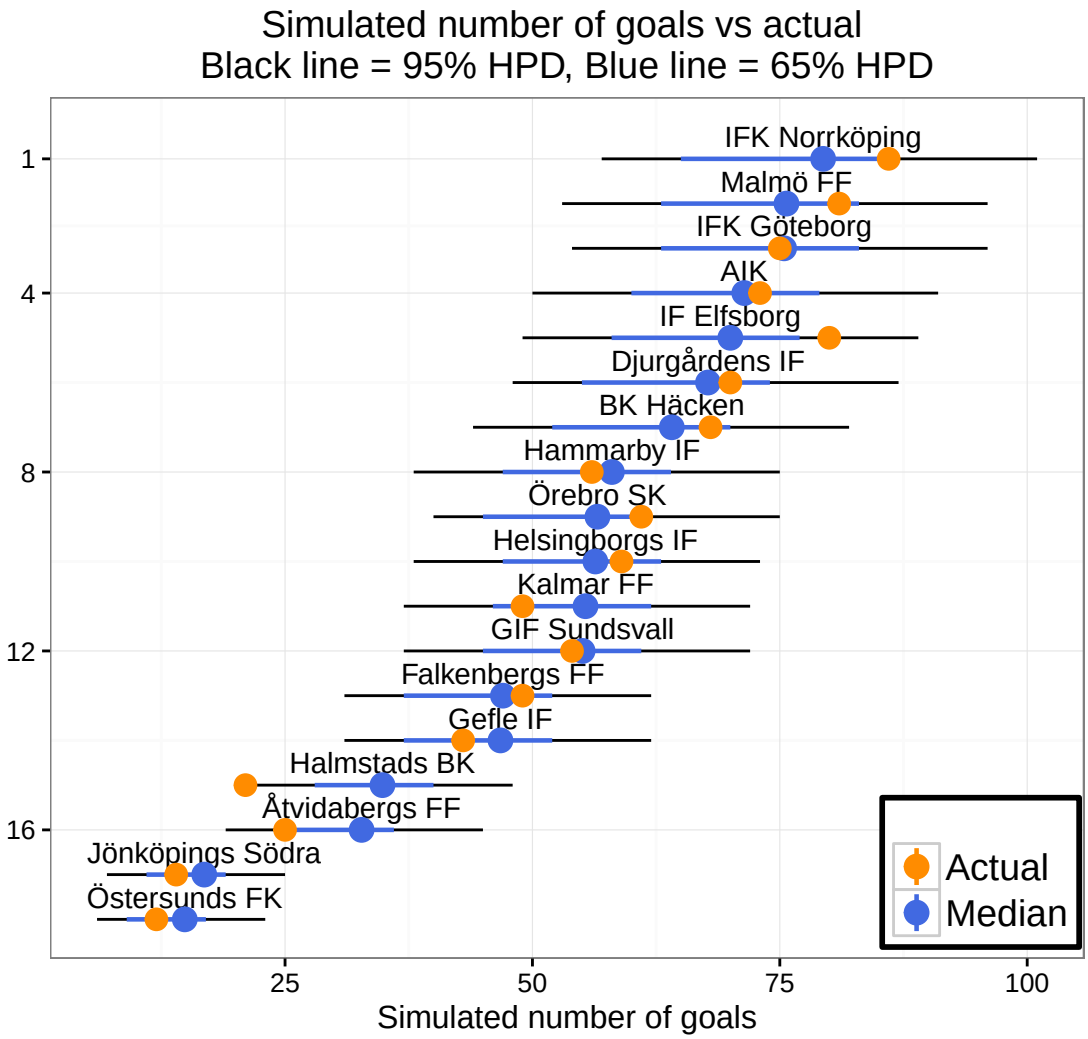
4.3.1 Predicted outcomes vs observed

The following graph compares the simulated number of points against the actual number of points won during the period. As before is both a 95 % and a 65 % HPD interval calculated.



The median of the simulated number of points for each team almost rank the teams in the same order as the actual points would have done. The big exception is Hammarby who are predicted to be four places higher than their actual number of points. At the same time can it be noted that Hammarbys actual number of points lies in the 65 % HPD interval and that the teams in the middle of Allsvenskan are very even.
For all estimated intervals can the actual number of points be found to lie either inside the 65 % interval or the 95 % interval. Although, a rather clear trend can also be seen as the median number of points always is lower than the actual number of points for the five top teams and in many cases higher than the actual points for the teams on the lower half.

In the next graph is the simulated number of goals compared against the observed number of goals.



Just as for the simulated number of points is there a trend for the simulated number of goals where the median for the top teams is lower than the actual number and the median for the teams in the bottom is higher than the actual number. Similar to before does also the simulations rank the teams in pretty much the right order and the intervals includes the actual number of goals in all cases but one. It is Halmstad whose actual number of goals is below the boundaries of the 95 % HPD interval.

4.3.2 Prediction of future outcomes

The remaining 144 games are predicted and put together with current table of the 2016 season. The column *Current* gives the points the teams have gathered so far this season and *Median* the median number of points during the rest of the season given the simulations. *Total* is the sum of the current points and the median number of points. The presented intervals are the sums of the current points plus lower or upper bound of the simulated numbe of points. Both a 95 % and a 65 % HPD interval is included.
The race for the title is predicted to be close with Malmö and Norrköping as main favorites and Göteborg together with AIK as possible contenders. Elfsborg could have a chance but their current might be a bit too low for them to really be able to challenge for the title. The middle of the table is really even and it does also look like it can become very close around place 14, the negative play-off. Gefle and Falkenberg are currently a bit behind and the model does not predict them to avoid relegation.

```
##                Team Current Median Total Lower95 Upper95 Lower65 Upper65
## 1          Malmö FF      27     32    59      46      69      52      63
## 2    IFK Norrköping      24     34    58      46      68      51      62
## 3      IFK Göteborg      21     32    53      41      63      48      58
## 4               AIK      22     30    52      39      62      45      56
## 5       IF Elfsborg      17     30    47      34      57      40      51
## 6         Örebro SK      23     23    46      34      56      40      50
## 7    Djurgårdens IF      15     28    43      32      55      37      48
## 8         BK Häcken      16     26    42      30      53      35      46
## 9     GIF Sundsvall      19     21    40      27      50      32      43
## 10 Jönköpings Södra      15     23    38      24      50      32      44
## 11  Helsingborgs IF      15     22    37      26      48      30      41
## 12        Kalmar FF      16     21    37      26      48      32      42
## 13      Hammarby IF      13     23    36      24      47      29      40
## 14    Östersunds FK      15     19    34      22      46      26      38
## 15         Gefle IF       5     17    22      11      32      15      25
## 16   Falkenbergs FF       4     17    21      10      31      14      24
```

The table above is a good indicator of which teams that are favorites to win the title and which teams that probably will be relegated. To obtain an even better view of these probabilities are the final positions saved for all of the 30 000 simulated seasons. The proportion of seasons a team wins the title, qualify for European tournaments (top 4), end up on the upper half, have to play negative play-off or is being relegated is presented in the table below.

Malmö won Allsvenskan in 42.2 % of the simulated season and was in the top 4 in approximately 9 out of 10 seasons. As mentioned is Norrköping the main rival and after them comes Göteborg and AIK who both won almost 10 % of the simulated seasons. Örebro, currently placed third in the table, are not predicted to be one of the title contenders with only a probability of 1.26 % to win the title. The relegation fight does seem to already be over as both Gefle and Falkenberg were relegated in about 85 % of the simulated seasons. Instead is it much more even in the fight for avoiding negative play-off. With a probability of 21.7 % is Östersund the current favorites but it will surely become a close race since five teams have a probability of 10 % or higher.

```
##                Team Winner  Top4  Top8 Rel_Play_off Relegation
## 1          Malmö FF  42.20 91.84 99.68         0.00       0.00
## 2    IFK Norrköping  35.52 90.16 99.39         0.01       0.00
## 3      IFK Göteborg   9.75 66.99 95.97         0.05       0.00
## 4               AIK   8.38 60.99 94.41         0.05       0.00
## 5       IF Elfsborg   1.64 29.66 80.29         0.66       0.07
## 6         Örebro SK   1.26 21.59 74.45         0.59       0.08
## 7    Djurgårdens IF   0.60 13.81 60.49         2.09       0.42
## 8         BK Häcken   0.29  9.14 52.12         3.13       0.59
## 9  Jönköpings Södra   0.16  4.63 30.29        11.10       3.79
## 10    GIF Sundsvall   0.12  4.64 36.45         5.97       1.22
## 11        Kalmar FF   0.04  2.07 22.40        10.55       3.06
## 12    Östersunds FK   0.03  1.26 13.60        21.70       9.52
## 13  Helsingborgs IF   0.02  1.75 22.18        12.07       3.30
## 14      Hammarby IF   0.00  1.48 18.10        14.82       5.58
## 15         Gefle IF   0.00  0.00  0.10         9.16      84.93
## 16   Falkenbergs FF   0.00  0.00  0.07         8.06      87.44
```

5 Discussion

The effect of playing at home have been proven to be significant for teams playing in La Liga(Bååth [2013]), Serie A(Baio and Blangiardo [2010]) and the Premier League(Weitzenfeld [2014]). In this report the advantage of playing at home was modeled by estimating one intercept for the home team and one for the away team. This approach was also used by Bååth[2013] and just as in his case the effect of playing at home was concluded to be significant. The log average goals scored per game for the home teams were estimated to lie in the 95 % HPD interval 0.38-0.55 and for the away teams the interval were 0.08-0.28. Hence, also for Allsvenskan it is concluded to be the right choice to include parameters in the model that estimates the home advantage.

Another property of the model that was of interest to investigate was whether it ranks the teams in a reasonable order. This was examined by looking at the expected number of goals when playing at home and when playing away. It can be concluded that the, by the model, estimated measure of team skill really well ranks the team. The top teams in the league (IFK Norrköping, Malmö, IFK Göteborg, AIK and Elfsborg) are the five highest ranked teams and after them follows Djurgården and Häcken. The middle of Allsvenskan is harder to rank since the teams are very even, which also is shown by the model as the estimates are very close. Among the teams with lowest number of expected goals are the teams who were relegated last year and the teams that currently lies in the relegation zone. The newcomers in the league has less observations compared to the other teams but the model still ranks them into a rather suitable position, and the intervals are only a little broader. Overall, the model produces a very reasonable ranking of the teams. These results are similar to, for instance, the ones produced by (Weitzenfeld [2014]) and confirms the belief on the general appropriateness of the model.

Both in the article by Baio and Blangiardo[2010] and the article by Weitzenfeld[2014] the problem of overshrinkage were mentioned. This is a problem also for the model presented in this report and is a result of setting the mean for the prior of the skill parameter to zero. The effect of this is that the skill for the top teams is underestimated and the skill of the bottom teams overestimated. The best and worst teams are both leaning a little bit too much towards the overall mean. However, it should also be mentioned that this not seem to have any severe negative effect on the results given by the model. The over- and underestimation is fairly low. A way to modify the model for dealing with this problem is proposed and tested by Baio and Blangiardo. The modified model is quite complicated but the general idea is to divide the teams into groups of for, instance, bottom, middle and top teams. For the respective groups are then the estimated skill parameters drawn from a truncated normal distribution and it is truncated differently depending on which group a team belongs to. This model resulted in less overshrinkage and could perhaps be an interesting extension to try out on the model presented in this report.

To summary, the hierarchical Bayesian model seems to work quite well for predicting Allsvenskan. Compared with the observed games did the predictions look promising and the prediction for the final standings of the 2016 season is, in my opinion, reasonable to say the least. However, if that is true or not, only time can tell.

6 References

Baio, Gianluca, & Blangiardo, Marta (2010). Bayesian hierarchical model for the prediction of football results. Journal Of Applied Statistics, 37(2), 253-264. doi:10.1080/02664760802684177

Bååth, Rasmus. Modeling Match Results in Soccer using a Hierarchical Bayesian Poisson Model. URL http://www.sumsar.net/papers/baath_2015_modeling_match_resluts_in_soccer.pdf. Accessed: 2016-06-02.

Gelman, Andrew, et al. Bayesian data analysis. Third edition. Boca Raton, FL, USA: Chapman & Hall/CRC, 2013.

Martyn Plummer (2016). rjags: Bayesian Graphical Models using MCMC. R package version 4-6. https://CRAN.R-project.org/package=rjags

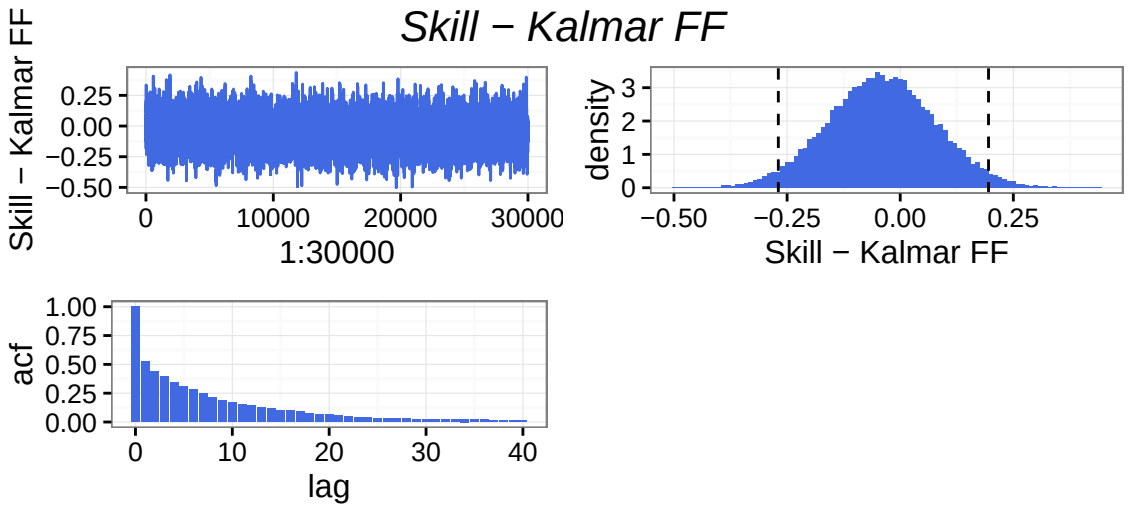
Martyn Plummer, Nicky Best, Kate Cowles and Karen Vines (2006). CODA: Convergence Diagnosis and Output Analysis for MCMC, R News, vol 6, 7-11

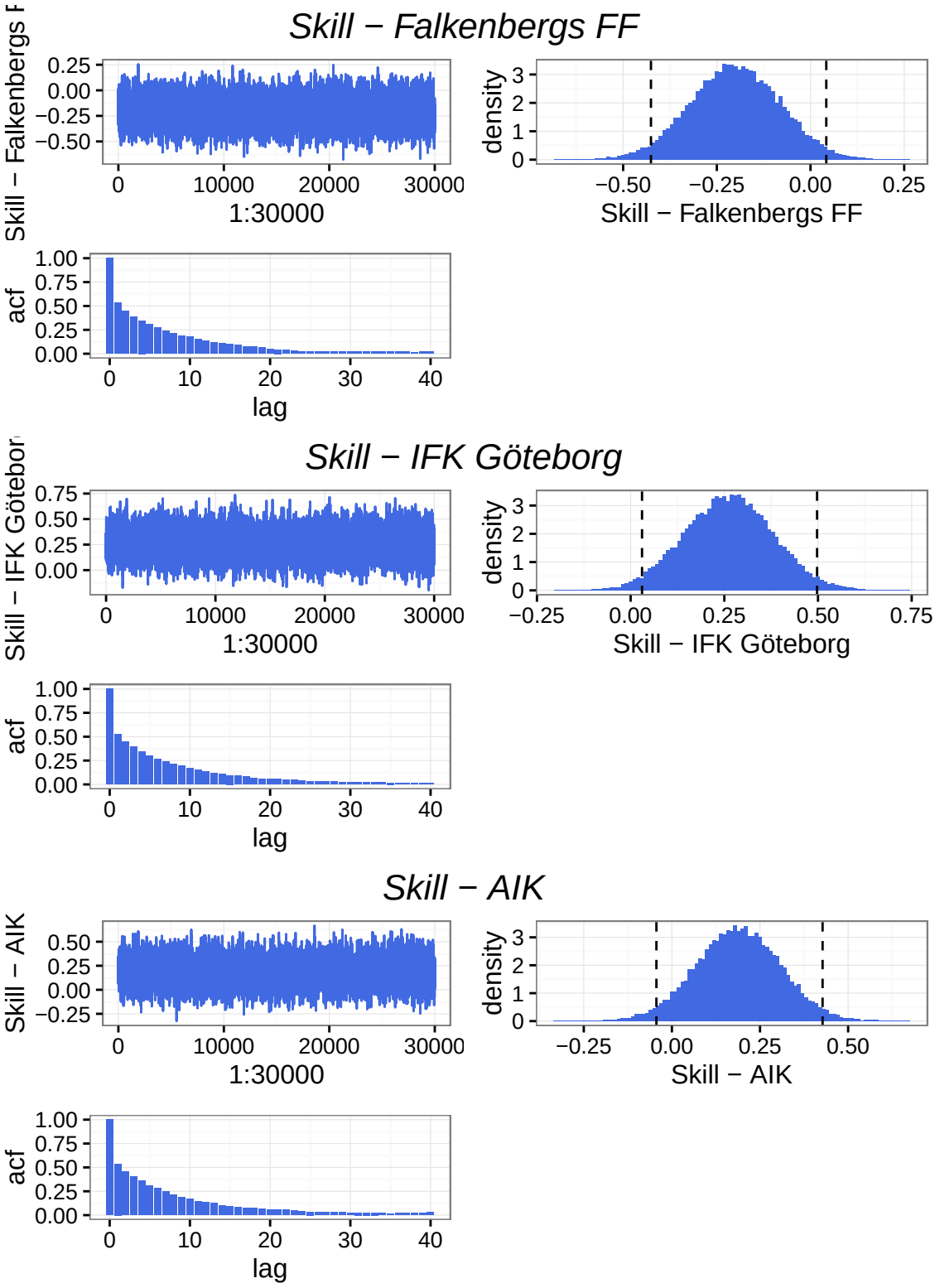
Weitzenfeld, Daniel. A Hierarchical Bayesian Model of the Premier League. URL http://danielweitzenfeld.github.io/passtheroc/blog/2014/10/28/bayes-premier-league/. Accessed: 2016-06-02.

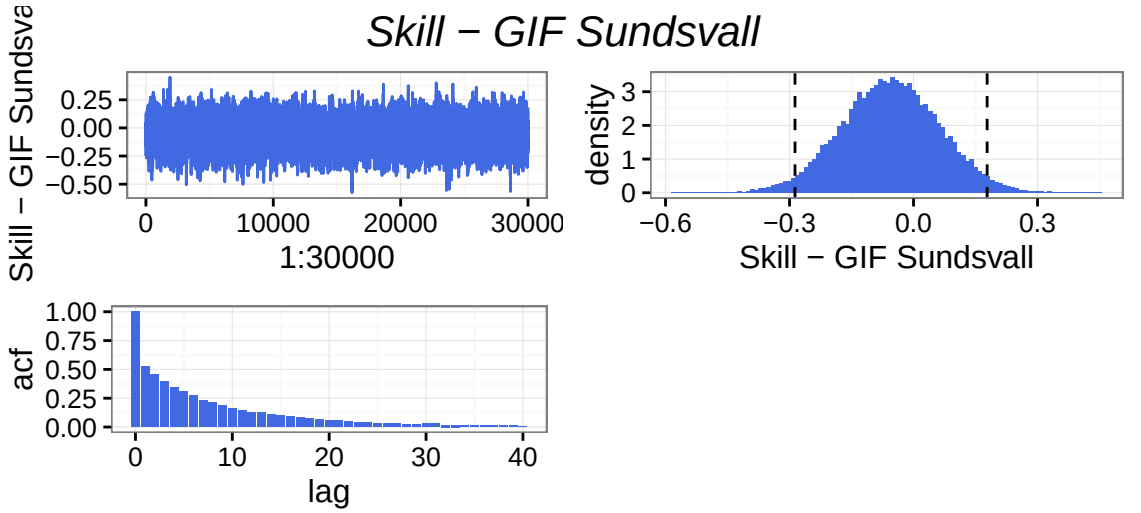
H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2009.

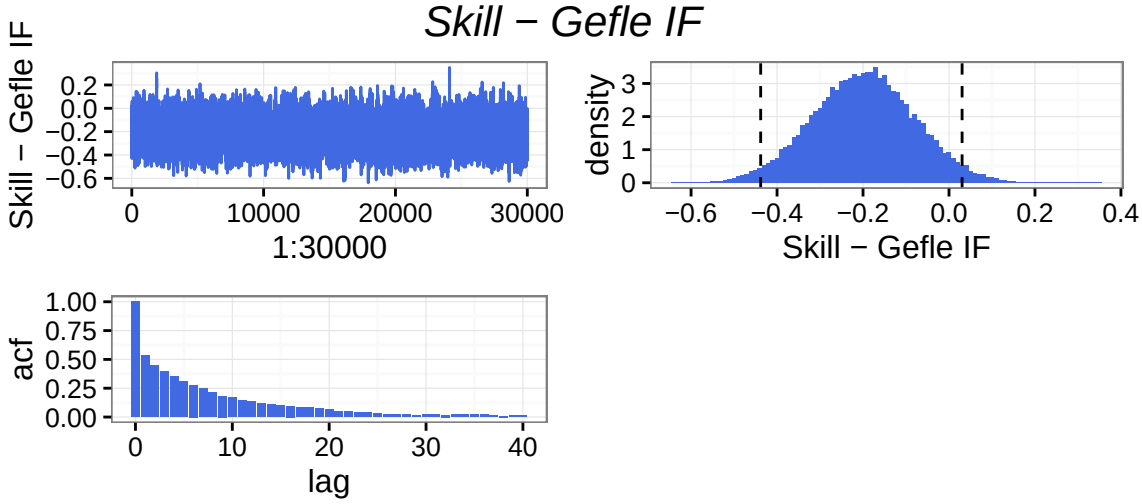
7 Appendix

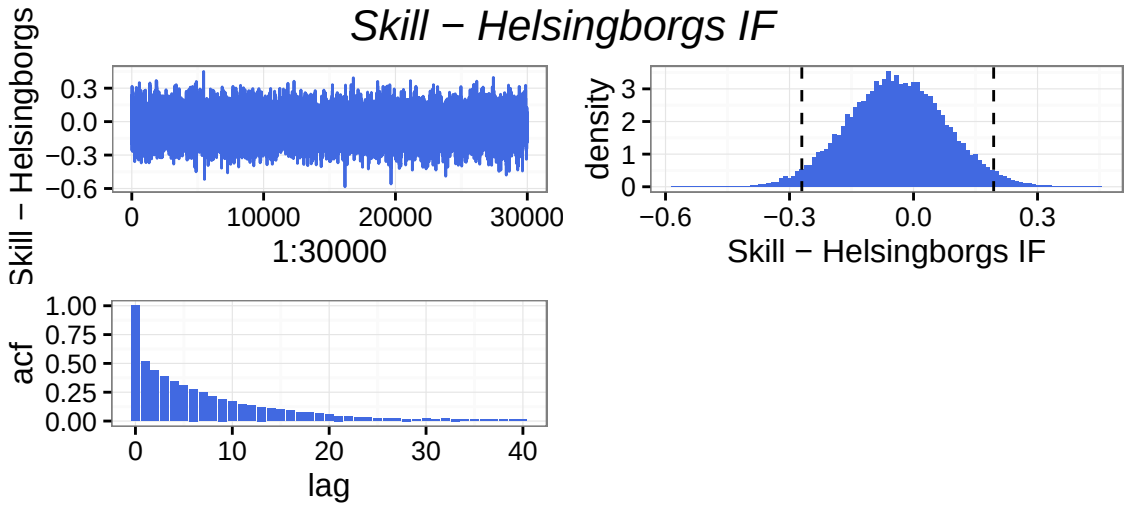
7.1 Convergence plots

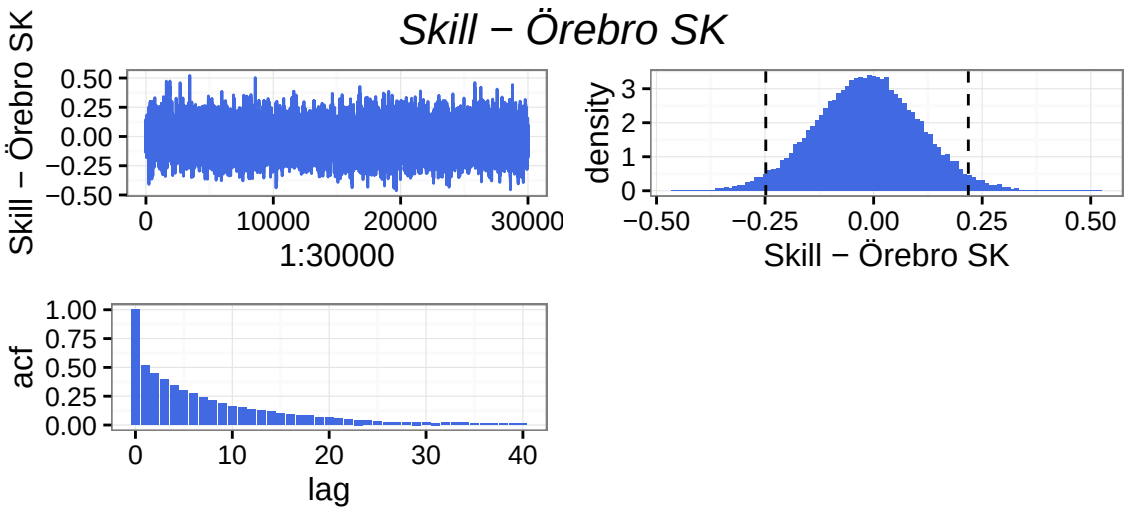


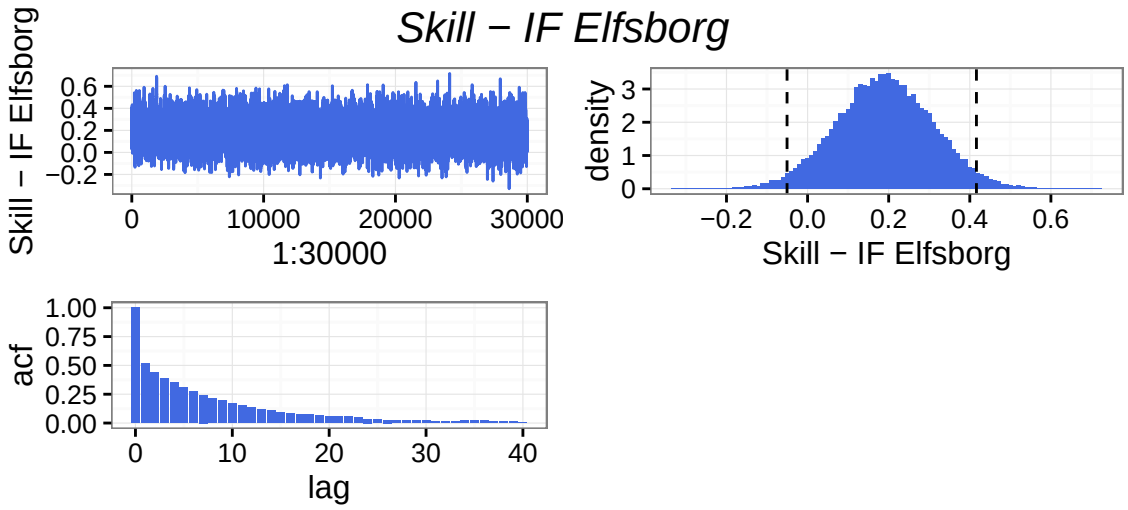

Skill − Falkenbergs FF
Skill − Falkenbergs F
0.25
0.00
−0.25
−0.50
0
10000
20000
30000
1:30000
density
3
2
1
0
−0.50
−0.25
0.00
0.25
Skill − Falkenbergs FF
acf
1.00
0.75
0.50
0.25
0.00
0
10
20
30
40
lag
Skill − IFK Göteborg
Skill − IFK Göteborg
0.75
0.50
0.25
0.00
0
10000
20000
30000
1:30000
density
3
2
1
0
−0.25
0.00
0.25
0.50
0.75
Skill − IFK Göteborg
acf
1.00
0.75
0.50
0.25
0.00
0
10
20
30
40
lag
Skill − AIK
Skill − AIK
0.50
0.25
0.00
−0.25
0
10000
20000
30000
1:30000
density
3
2
1
0
−0.25
0.00
0.25
0.50
Skill − AIK
acf
1.00
0.75
0.50
0.25
0.00
0
10
20
30
40
lag

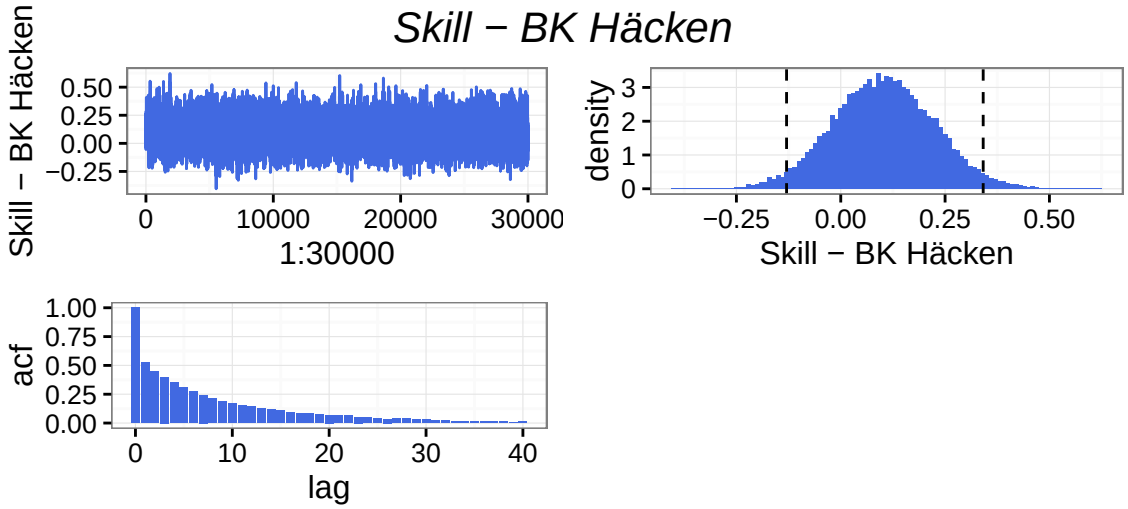

Skill − GIF Sundsvall
Skill − GIF Sundsva
0.25
0.00
−0.25
−0.50
0
10000
20000
30000
1:30000
density
3
2
1
0
−0.6
−0.3
0.0
0.3
Skill − GIF Sundsvall
acf
1.00
0.75
0.50
0.25
0.00
0
10
20
30
40
lag

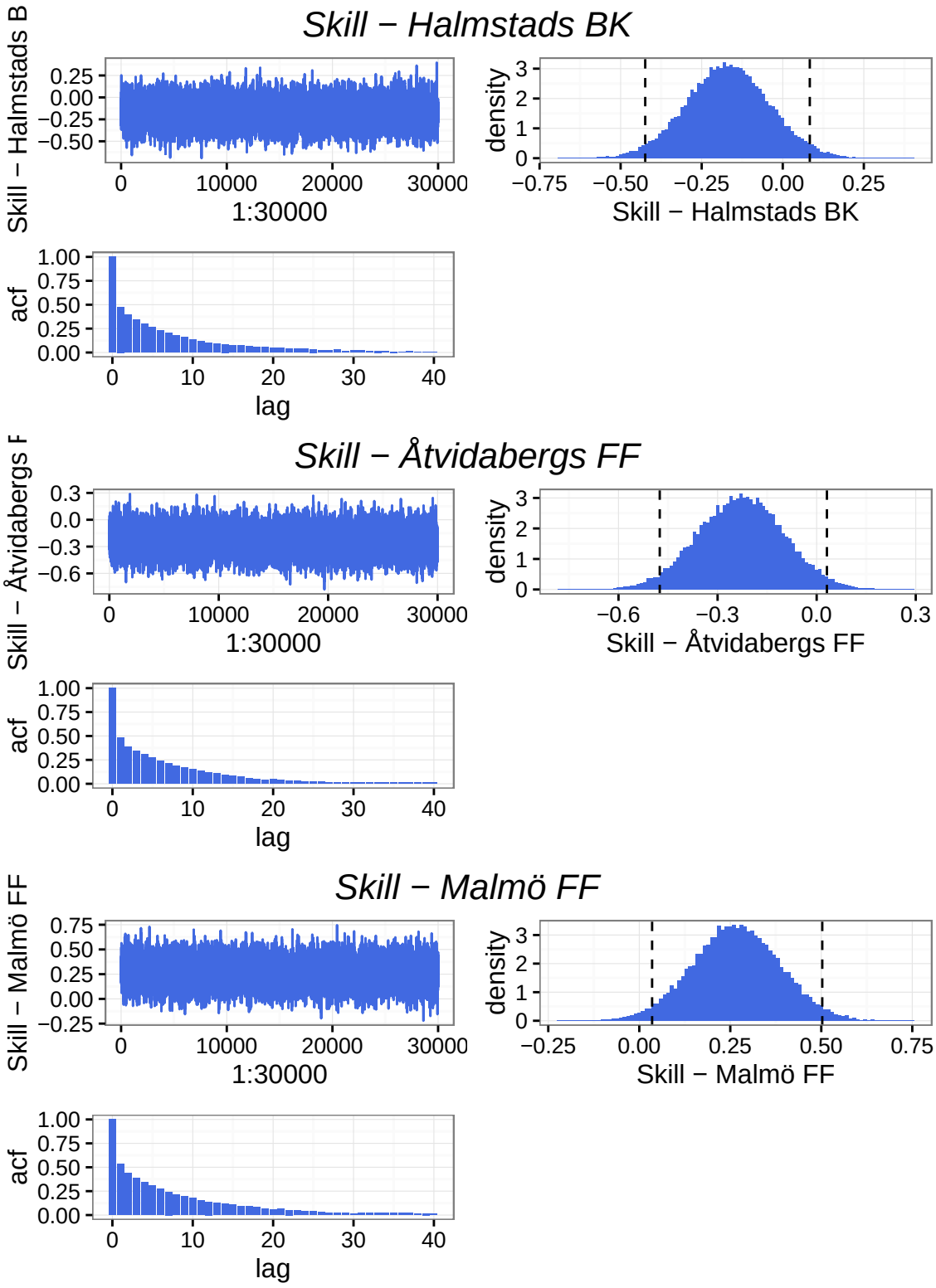

Skill − Gefle IF
Skill − Gefle IF
0.2
0.0
−0.2
−0.4
−0.6
0
10000
20000
30000
1:30000
density
3
2
1
0
−0.6
−0.4
−0.2
0.0
0.2
0.4
Skill − Gefle IF
acf
1.00
0.75
0.50
0.25
0.00
0
10
20
30
40
lag

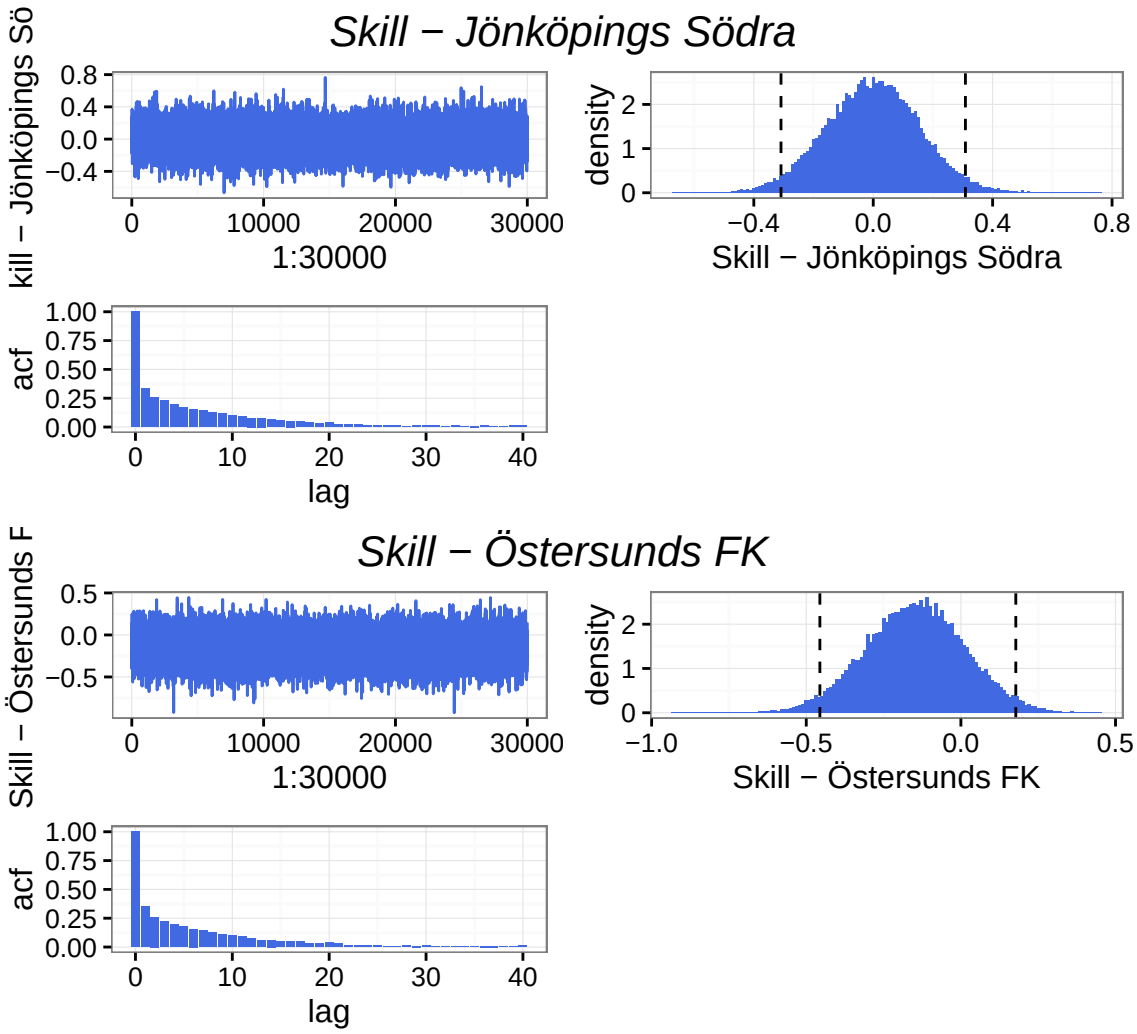

Skill − Helsingborgs IF
Skill − Helsingborgs
0.3
0.0
−0.3
−0.6
0
10000
20000
30000
1:30000
density
3
2
1
0
−0.6
−0.3
0.0
0.3
Skill − Helsingborgs IF
acf
1.00
0.75
0.50
0.25
0.00
0
10
20
30
40
lag


Skill − Örebro SK
Skill − Örebro SK
0.50
0.25
0.00
−0.25
−0.50
0
10000
20000
30000
1:30000
density
3
2
1
0
−0.50
−0.25
0.00
0.25
0.50
Skill − Örebro SK
acf
1.00
0.75
0.50
0.25
0.00
0
10
20
30
40
lag


Skill − IF Elfsborg
Skill − IF Elfsborg
0.6
0.4
0.2
0.0
−0.2
0
10000
20000
30000
1:30000
density
3
2
1
0
−0.2
0.0
0.2
0.4
0.6
Skill − IF Elfsborg
acf
1.00
0.75
0.50
0.25
0.00
0
10
20
30
40
lag


Skill − BK Häcken
Skill − BK Häcken
0.50
0.25
0.00
−0.25
0
10000
20000
30000
1:30000
density
3
2
1
0
−0.25
0.00
0.25
0.50
Skill − BK Häcken
acf
1.00
0.75
0.50
0.25
0.00
0
10
20
30
40
lag


Skill − Halmstads BK
Skill − Halmstads B
0.25
0.00
−0.25
−0.50
0
10000
20000
30000
1:30000
density
3
2
1
0
−0.75
−0.50
−0.25
0.00
0.25
Skill − Halmstads BK
acf
1.00
0.75
0.50
0.25
0.00
0
10
20
30
40
lag
Skill − Åtvidabergs FF
Skill − Åtvidabergs F
0.3
0.0
−0.3
−0.6
0
10000
20000
30000
1:30000
density
3
2
1
0
−0.6
−0.3
0.0
0.3
Skill − Åtvidabergs FF
acf
1.00
0.75
0.50
0.25
0.00
0
10
20
30
40
lag
Skill − Malmö FF
Skill − Malmö FF
0.75
0.50
0.25
0.00
−0.25
0
10000
20000
30000
1:30000
density
3
2
1
0
−0.25
0.00
0.25
0.50
0.75
Skill − Malmö FF
acf
1.00
0.75
0.50
0.25
0.00
0
10
20
30
40
lag



7.2 R-code

```
# 1. Read in libraries
library(plyr); library(dplyr)
library(rjags)
library(coda)
library(stringr)
library(ggplot2)
library(grid)
library(gridExtra)

# 2. Read in data
res2015 <- read.csv("C:\\Users\\Gustav\\Documents\\Bayesian-Learning\\Project\\Allsvenskan2015.csv",
                    sep=",", header = TRUE)
res2015 <- res2015[,c(21,5,14,23)]
names(res2015) <- c("Home.Team",  "Away.Team", "Home.Goals", "Away.Goals")
res2016 <- read.csv("C:\\Users\\Gustav\\Documents\\Bayesian-Learning\\Project\\Allsv2016.csv",
                    sep=";", header = TRUE)
```

```
res1516 <- (bind_rows(res2015, res2016[1:96,]))

# 3. Specify model
## First, a list with information from the data frame
teams <- unique(c(res1516$Home.Team, res1516$Away.Team))

Allsv_list <- list(HomeGoals=res1516$Home.Goals,AwayGoals=res1516$Away.Goals,
                   HomeTeam=as.numeric(factor(res1516$Home.Team, levels=teams)),
                   AwayTeam=as.numeric(factor(res1516$Away.Team, levels=teams)),
                teams = length(teams), games = nrow(res1516))

## Second, write a string with the model for the JAGS function
MyModelString <- "model {
  for (i in 1:games){
    HomeGoals[i] ~ dpois(lambdaHome[HomeTeam[i],AwayTeam[i]])
    AwayGoals[i] ~ dpois(lambdaAway[HomeTeam[i],AwayTeam[i]])
  }
  for(home in 1:teams){
    for(away in 1:teams){
      lambdaHome[home, away] <- exp(homeInt + skill[home] - skill[away])
      lambdaAway[home, away] <- exp(awayInt + skill[away] - skill[home])
    }
  }
skill[1] <- 0
for(j in 2:teams) {
  skill[j] ~ dnorm(group_mu, group_precision)
}
group_mu ~ dnorm(0, 0.0625)
group_precision <- 1 / pow(group_sigma, 2)
group_sigma ~ dunif(0, 3)
homeInt ~ dnorm(0, 0.0625)
awayInt ~ dnorm(0, 0.0625)
} "
# 4. Compile model and generate MCMC samples
## Compiling model
AllsvModel <- jags.model(textConnection(MyModelString), data=Allsv_list, n.chains=3, n.adapt=10000)
## Burning some samples
update(AllsvModel, 10000)
## Generate MCMC samples
s1 <- coda.samples(AllsvModel, variable.names=c("homeInt", "awayInt", "skill", "group_mu",
                                                "group_sigma"), n.iter=20000, thin=2)
## Merge the MCMC chains into one matrix
ms1 <- as.matrix(s1)
ms1df <- as.data.frame(ms1)
# 5. Analyze convergence of the parameters
conv_plots <- function(indexParam, nameParam){
  trace <- ggplot(ms1df, aes(x=1:30000, y=ms1df[,indexParam])) + geom_line(col="royalblue") +
    theme_bw() + ylab(nameParam)
  HPD_lower <- HPDinterval(as.mcmc(ms1[,indexParam]), prob=0.95)[1]
  HPD_upper <- HPDinterval(as.mcmc(ms1[,indexParam]), prob=0.95)[2]
  density <- ggplot(ms1df, aes(x=ms1df[,indexParam])) + geom_density() + theme_bw() +
    geom_vline(xintercept=c(HPD_lower, HPD_upper), linetype=2)
  hist <- ggplot(ms1df, aes(x=ms1df[,indexParam])) + geom_histogram(aes(y=..density..),fill="royalblue"
```

```
                                                                    binwidth=0.01) + theme_bw() +
    geom_vline(xintercept=c(HPD_lower, HPD_upper), linetype=2) + xlab(nameParam)
  ACF <- data.frame(acf=as.numeric(acf(ms1df[,indexParam], plot = FALSE, lag.max = 40)$acf), lag=0:40)
  Efficiency <- ggplot(ACF, aes(x=lag, y=acf)) + geom_bar(fill="royalblue", stat="identity") + theme_bw
  text <- nameParam
  plots <- grid.arrange(trace, hist, Efficiency, ncol=2, top=textGrob(text,gp=gpar(fontsize=20,font=3)))
  return(plots)
}

awInt <- conv_plots(1, "Away intercept")
sigma <- conv_plots(2, "Sigma - Teams")
skill <- conv_plots(3, "Mu - Teams")
hmInt <- conv_plots(4, "Home intercept")
Peking <- conv_plots(10, "Skill - IFK Norrköping")

# 6. Compare estimates of the skill parameters for the teams, home and away
## Home
teamSkill <- ms1[,5:22]
teamSkillHome <- (teamSkill - rowMeans(teamSkill)) + ms1[, "homeInt"]
teamSkillHome <- exp(teamSkillHome)
colnames(teamSkillHome) <- teams
teamSkillHome <- teamSkillHome[,order(colMeans(teamSkillHome), decreasing=T)]

teamSkillHome_frame <- data.frame(HPDinterval(as.mcmc(teamSkillHome), prob=0.95), median =
                                    as.numeric(apply(teamSkillHome, 2, FUN = median)) )
teamSkillHome_frame$team <- row.names(teamSkillHome_frame)
teamSkillHome_frame[,5:6] <-  HPDinterval(as.mcmc(teamSkillHome), prob=0.65)
ggplot(teamSkillHome_frame, aes(y=median, x=1:18)) + theme_bw() +
  geom_pointrange(aes(ymin = lower, ymax = upper),col="royalblue") +
  coord_flip() + geom_text(aes(label=team),nudge_x = 0.5) + geom_pointrange(aes(ymin = V5, ymax =  V6),
                                                                    size=1.05,col="royalblue") +
  scale_x_reverse(breaks=c(1,4,8,12,16))

## Away
teamSkillAway <- (teamSkill - rowMeans(teamSkill)) + ms1[, "awayInt"]
teamSkillAway <- exp(teamSkillAway)
colnames(teamSkillAway) <- teams
teamSkillAway <- teamSkillAway[,order(colMeans(teamSkillAway), decreasing=T)]

teamSkillAway_frame <- data.frame(HPDinterval(as.mcmc(teamSkillAway), prob=0.95), median =
                                    as.numeric(apply(teamSkillAway, 2, FUN = median)) )
teamSkillAway_frame$team <- row.names(teamSkillAway_frame)
teamSkillAway_frame[,5:6] <-  HPDinterval(as.mcmc(teamSkillAway), prob=0.65)
ggplot(teamSkillAway_frame, aes(y=median, x=1:18)) + theme_bw() + geom_pointrange(aes(ymin = lower,
                                                                    ymax =upper),col="royalblue") +
  coord_flip() + geom_text(aes(label=team),nudge_x = 0.5) + geom_pointrange(aes(ymin = V5, ymax =  V6),
                                                                    size=1.05,col="royalblue") +
  scale_x_reverse(breaks=c(1,4,8,12,16))

# 7. Predictions
Allres1516 <- (bind_rows(res2015, res2016))
PredGoalsHome <-  data.frame(matrix(vector(), 30000, 480))
PredGoalsAway <-  data.frame(matrix(vector(), 30000, 480))
```

```
n <- nrow(ms1)
for(i in 1:nrow(Allres1516)){
  home_team <- which(teams == Allres1516$Home.Team[i])
  away_team <- which(teams == Allres1516$Away.Team[i])
  home_skill <- ms1[, home_team + 4]
  away_skill <- ms1[, away_team + 4]
  homeInt <- ms1[, 4]
  awayInt <- ms1[, 1]
  home_goals <- rpois(n, exp(homeInt + home_skill - away_skill))
  PredGoalsHome[,i] <- home_goals
  away_goals <- rpois(n, exp(awayInt + away_skill - home_skill))
  PredGoalsAway[,i] <- away_goals
}
# Simulated match results
{
res1516$MatchResult <- sign(res1516$Home.Goals - res1516$Away.Goals)

MatchResults <- data.frame(matrix(vector(), 30000, 480))
for(i in 1:480){
  MatchResults[,i] <- sign(PredGoalsHome[,i] - PredGoalsAway[,i])
}
MatchResults <- data.frame(t(MatchResults))

AllRes <- cbind(Allres1516[1:336,1:2],MatchResults[1:336,])
seasonP <- data.frame(matrix(vector(), 18, 30001))
seasonP[,1] <- data.frame(table(AllRes$Home.Team))[,1]
for(i in 2:30001){
  seasonP[,i] <- as.numeric(table(AllRes$Home.Team, AllRes[,i+1])[,3] *3) +
    as.numeric(table(AllRes$Home.Team, AllRes[,i+1])[,2]) +
    as.numeric(table(AllRes$Away.Team, AllRes[,i+1])[,1] *3) + as.numeric(table(AllRes$Away.Team,
                                                                                AllRes[,i+1])[,2])
}
seasonP <- t(seasonP)
colnames(seasonP) = seasonP[1, ] # the first row will be the header
seasonP = seasonP[-1, ]          # removing the first row.

seasonInt <- data.frame(matrix(vector(), 18, 6))
for(i in 1:18){
  seasonInt[i,2:3] <- data.frame(HPDinterval(as.mcmc(as.numeric(seasonP[,i])), prob=0.95) )
  seasonInt[i,4:6] <- data.frame(HPDinterval(as.mcmc(as.numeric(seasonP[,i])), prob=0.65),
                                 median=as.numeric(apply(data.frame(as.numeric(seasonP[,i])), 2,
                                                         FUN = median)) )
}
seasonInt[,1] <- data.frame(table(AllRes$Home.Team))[,1]
names(seasonInt) <- c("Team", "Lower95", "Upper95", "Lower65","Upper65", "Median")

seasonInt$Actual <- as.numeric(table(res1516$Home.Team,res1516$MatchResult)[,3] *3) +
  as.numeric(table(res1516$Home.Team, res1516$MatchResult)[,2]) +
  as.numeric(table(res1516$Away.Team, res1516$MatchResult)[,1] *3) + as.numeric(table(res1516$Away.Team
                                                                                     res1516$MatchResul

seasonInt <-seasonInt[order(-seasonInt$Median),]
cols <- c("Median"="royalblue","Actual"="darkorange")
```

```r
ggplot(seasonInt, aes(y=Median,x=1:18))+scale_x_reverse(breaks=c(1,4,8,12,16))+theme_bw()+coord_flip()+
  geom_text(aes(label=Team),nudge_x=0.5)+ geom_pointrange(aes(ymin=Lower95,ymax=Upper95),col="black")+
  geom_pointrange(aes(ymin=Lower65,ymax=Upper65,col="Median"), size=0.8) +
  geom_point(aes(y=Actual,col="Actual"), size=3.5) + scale_colour_manual(name="",values=cols) +
  theme(legend.position=c(.9, .1),legend.background = element_rect(color = "black",   fill = "white", si
                                                                   = 1, linetype = "solid"), legend.text
          element_text(size = 14)) + ggtitle("Simulated number of points vs actual \nBlack line = 95% HP
                                             Blue line = 65% HPD") + ylab("Simulated number of points")
}

# Simulated number of goals

HomeGoals <- cbind(Allres1516[1:336,1:2],t(PredGoalsHome)[1:336,])
AwayGoals <- cbind(Allres1516[1:336,1:2],t(PredGoalsAway)[1:336,])

SeasonGoals <- data.frame(matrix(vector(), 18, 30001))
SeasonGoals[,1] <- data.frame(table(AllRes$Home.Team))[,1]
for(i in 2:30001){
  SeasonGoals[,i] <- aggregate(HomeGoals[,i+1], by=list(Team=HomeGoals$Home.Team), FUN=sum)[,2] +
    aggregate(AwayGoals[,i+1], by=list(Team=AwayGoals$Away.Team), FUN=sum)[,2]
}
SeasonGoals <- t(SeasonGoals)
colnames(SeasonGoals) = SeasonGoals[1, ] # the first row will be the header
SeasonGoals = SeasonGoals[-1, ]          # removing the first row.

seasonGoalInt <- data.frame(matrix(vector(), 18, 6))
for(i in 1:18){
  seasonGoalInt[i,2:3] <- data.frame(HPDinterval(as.mcmc(as.numeric(SeasonGoals[,i])), prob=0.95) )
  seasonGoalInt[i,4:6] <- data.frame(HPDinterval(as.mcmc(as.numeric(SeasonGoals[,i])), prob=0.65),
                                     median=as.numeric(apply(data.frame(as.numeric(SeasonGoals[,i])), 2, FU
}
seasonGoalInt[,1] <- data.frame(table(AllRes$Home.Team))[,1]
names(seasonGoalInt) <- c("Team", "Lower95", "Upper95", "Lower65","Upper65", "Median")

seasonGoalInt$Actual <- aggregate(res1516$Home.Goals, by=list(Team=res1516$Home.Team), FUN=sum)[,2] +
  aggregate(res1516$Away.Goals, by=list(Team=res1516$Away.Team), FUN=sum)[,2]

seasonGoalInt <-seasonGoalInt[order(-seasonGoalInt$Median),]
cols <- c("Median"="royalblue","Actual"="darkorange")
ggplot(seasonGoalInt, aes(y=Median,x=1:18))+scale_x_reverse(breaks=c(1,4,8,12,16))+theme_bw()+coord_flip
  geom_text(aes(label=Team),nudge_x=0.5)+ geom_pointrange(aes(ymin=Lower95,ymax=Upper95),col="black")+
  geom_pointrange(aes(ymin=Lower65,ymax=Upper65,col="Median"), size=0.8) +
  geom_point(aes(y=Actual,col="Actual"), size=3.5) + scale_colour_manual(name="",values=cols) +
  theme(legend.position=c(.9, .1),legend.background = element_rect(color = "black",   fill = "white", si
                                                                   = 1, linetype = "solid"),legend.text
          element_text(size = 14)) +  ggtitle("Simulated number of goals vs actual \nBlack line = 95% HP
                                              Blue line = 65% HPD")+ ylab("Simulated number of goals") +

# Future outcomes

res2016$MatchResult <- sign(res2016$Home.Goals - res2016$Away.Goals)
AllFut <- cbind(Allres1516[337:480,1:2],MatchResults[337:480,])
seasonFut <- data.frame(matrix(vector(), 16, 30001))
```

```
seasonFut[,1] <- data.frame(table(AllFut$Home.Team))[,1]
for(i in 2:30001){
  seasonFut[,i] <- as.numeric(table(AllFut$Home.Team, AllFut[,i+1])[,3] *3) + as.numeric(table(AllFut$H
    as.numeric(table(AllFut$Away.Team, AllFut[,i+1])[,1] *3) + as.numeric(table(AllFut$Away.Team, AllFu
}
seasonFut <- t(seasonFut)
colnames(seasonFut) = seasonFut[1, ] # the first row will be the header
seasonFut = seasonFut[-1, ]          # removing the first row.

seasonPred <- data.frame(matrix(vector(), 16, 6))
for(i in 1:16){
  seasonPred[i,2:3] <- data.frame(HPDinterval(as.mcmc(as.numeric(seasonFut[,i])), prob=0.95) )
  seasonPred[i,4:6] <- data.frame(HPDinterval(as.mcmc(as.numeric(seasonFut[,i])), prob=0.65),
                                  median=as.numeric(apply(data.frame(as.numeric(seasonFut[,i])), 2, FUN =
}
seasonPred[,1] <- data.frame(table(AllFut$Home.Team))[,1]
names(seasonPred) <- c("Team", "Lower95", "Upper95", "Lower65","Upper65", "Median")
seasonPred$Current <- as.numeric(table(res2016$Home.Team,res2016$MatchResult)[,3] *3) + as.numeric(table
  as.numeric(table(res2016$Away.Team, res2016$MatchResult)[,1] *3) +
  as.numeric(table(res2016$Away.Team, res2016$MatchResult)[,2])
seasonPred$Total <- seasonPred$Median + seasonPred$Current
Current = seasonPred$Current
seasonPred <-data.frame(Team=data.frame(table(AllFut$Home.Team))[,1],Current=seasonPred$Current,
                        Median=seasonPred$Median, Total=seasonPred$Total, Lower95=seasonPred$Lower95 +
                          Current,  Upper95=seasonPred$Upper95 + Current,Lower65=seasonPred$Lower65 +
                          Current,Upper65=seasonPred$Upper65 + Current)
seasonPred <-seasonPred[order(-seasonPred$Total),]
row.names(seasonPred) <- NULL
seasonPred

seasonFut2 <- t(seasonFut)
# Add current points
for(i in 1:16){
  seasonFut2[i,] <- as.numeric(seasonFut2[i,]) + seasonPred[i,7]
}

SeasonPos <- data.frame(matrix(vector(), 16, 30001))
SeasonPos[,1] <- data.frame(table(AllFut$Home.Team))[,1]
for(i in 2:30001){
  x <- as.numeric(seasonFut2[,i-1])
  SeasonPos[,i] <- rank(-x, ties.method = "random")
}
SeasonPos <- t(SeasonPos)
colnames(SeasonPos) = SeasonPos[1, ] # the first row will be the header
SeasonPos = SeasonPos[-1, ]          # removing the first row.

prob <- data.frame(Team=data.frame(table(AllFut$Home.Team))[,1],Winner=0, Top4=0,
                   Top8=0, Rel_Play_off=0, Relegation=0)
Win <- 1
Top4 <- seq(1,4,1)
Top8 <- seq(1,8,1)
Rel_Play_off <- 14
Relegation <- seq(15,16,1)
```

```
for(i in 1:16){
  prob[i,2] <- round(mean(as.numeric(SeasonPos[,i]) %in% Win) * 100,2)
  prob[i,3]  <-round(mean(as.numeric(SeasonPos[,i]) %in% Top4) * 100,2)
  prob[i,4] <- round(mean(as.numeric(SeasonPos[,i]) %in% Top8) * 100,2)
  prob[i,5]  <- round(mean(as.numeric(SeasonPos[,i]) %in% Rel_Play_off) * 100,2)
  prob[i,6]  <- round(mean(as.numeric(SeasonPos[,i]) %in% Relegation) * 100,2)
}
prob <-prob[order(-prob$Winner,-prob$Top4,-prob$Top8,-prob$Rel_Play_off,-prob$Relegation),]
row.names(prob) <- NULL
prob

skill <- conv_plots(6, "Skill - Kalmar FF")
skill <- conv_plots(7, "Skill - Falkenbergs FF")
skill <- conv_plots(8, "Skill - IFK Göteborg")
skill <- conv_plots(11, "Skill - AIK")
skill <- conv_plots(12, "Skill - GIF Sundsvall")
skill <- conv_plots(13, "Skill - Gefle IF")
skill <- conv_plots(14, "Skill - Helsingborgs IF")
skill <- conv_plots(15, "Skill - Örebro SK")
skill <- conv_plots(16, "Skill - IF Elfsborg")
skill <- conv_plots(17, "Skill - BK Häcken")
skill <- conv_plots(18, "Skill - Halmstads BK")
skill <- conv_plots(19, "Skill - Åtvidabergs FF")
skill <- conv_plots(20, "Skill - Malmö FF")
skill <- conv_plots(21, "Skill - Jönköpings Södra")
skill <- conv_plots(22, "Skill - Östersunds FK")
```