Final Project - Statistical Methods in Data Science II

Bayesian hierarchical models for the prediction of football results

Davide Cacciatore - 2015641

Contents

1 Introduction

Statistical modelling of sports data is a popular topic and much research has been produced for this purpose, also with reference to football.

For this project, we considered the work done by Baio & Blangiardo (2010) to model football results during a Serie A season. We attempted to apply their first model, assuming two conditionally independent Poisson variables for the number of goals scored. Next, we will attempt to compare its performance with a Negative Binomial model proposed in past work (Pollard et al. 1977) and with a frequentist approach. We applied these models to the 2021-2022 Italian Serie A data.

2 Data

We took data for the Italian Serie A 2021-2022 from a Kaggle dataset called *Football Data from Transfermarkt*, which contains structured and automatically updated football data scraped from the Transfermarkt website on all the European matches. We used MySQL Workbench to execute an SQL query, available in the Appendix, to extract only data from the last Serie A season.

2.1 Preprocessing

We had to perform several preprocessing steps on the dataset, in order to obtain the same data structure as the paper.

- We noticed that three matches were missing in the dataset and we inserted them manually.
- We replaced the column with the 'matchdays' with an increasing number from 1 to 38 indicating the Serie A round.
- We added an ID for each match, an increasing number from 1 to 380 identifying the individual match played.
- We renamed the team names legibly.
- We assigned, in alphabetical order, some indices uniquely associated with the 20 teams.

In the following tables, we can see the final structure of the data.

```
head(games)
```

```
##   g home.team   away.team h.g a.g Yg1 Yg2
## 1 1  Cagliari      Spezia   3  17   2   2
## 2 2 Sampdoria       Milan  15  11   0   1
## 3 3    Napoli     Venezia  12  20   2   0
## 4 4     Inter       Genoa   8   6   4   0
## 5 5   Bologna Salernitana   2  14   3   2
## 6 6   Udinese    Juventus  19   9   2   2
```

```
tail(games)
```

```
##       g   home.team away.team h.g a.g Yg1 Yg2
## 375 375      Spezia    Napoli  17  12   0   3
## 376 376       Genoa   Bologna   6   2   0   1
## 377 377 Salernitana   Udinese  14  19   0   4
## 378 378  Fiorentina  Juventus   5   9   2   0
## 379 379    Atalanta    Empoli   1   4   0   1
## 380 380      Torino      Roma  18  13   0   3
```

It consists of the match code *g*, the name of the teams, the number of goals scored per match (*Yg1* and *Yg2*) by the two teams, and the indexes *h.g* and *a.g* uniquely associated with each team. For example, Napoli is always associated with index 12, whether it plays away, as in *a(375)*, or at home, as in *h(3)*.

2.2 Exploratory Data Analysis

We can study the summary of the variable of goals scored at home (*Yg1*) and the variable of goal scored away (*Yg2*).

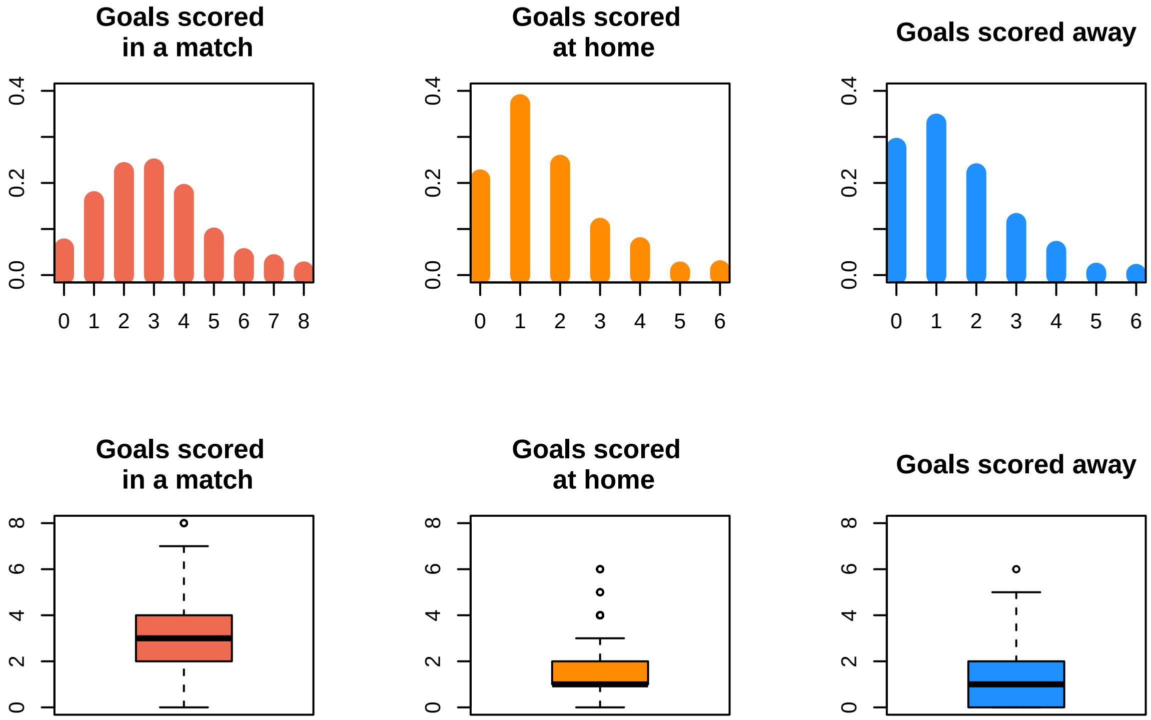
```
summary(games$Yg1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   1.000   1.503   2.000   6.000
```

```
summary(games$Yg2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   1.000   1.363   2.000   6.000
```

We can observe that the average number of goals scored at home is slightly higher than the average number of goals scored away. This is an expected result, because it is a well-known phenomenon: the home team has a small advantage over its opponent. Later, this small difference will be taken into account in the structure of the model. Let us look at the plots of these distributions.



Obviously, we can see the differences between the three results. On the left, we found the distribution of total goals scored in a match, in the middle we have the goals scored at home and on the right the goals scored away. Looking at the shape of the distributions and their discrete nature, it is clear that a good choice for the known distribution in our model could be a Poisson or a Negative Binomial.

3. Bayesian Hierarchical Model

The first model we decided to implement is the one described by Baio & Blangiardo. The Italian Serie A league consists of a total of 20 teams, which face each other twice in a season (once home and once away). The number of goals scored by the home and away teams in the g-th game of the season (g=1, ..., 380) are indicated by the variables *Yg1* and *Yg2*. The two distributions can be modelled as two conditionally independent Poissons.

$$\begin{aligned} Y_{g1} &\sim \text{Poisson}(\theta_{g1}) \\ Y_{g2} &\sim \text{Poisson}(\theta_{g2}) \end{aligned}$$

The parameters θ_{g1} and θ_{g2} represent the scoring intensity in the g-th game for the home and away team, respectively. These parameters are modelled according to a formulation widely used in the statistical literature (Karlis & Ntzoufras 2003), assuming a log-linear random effects model:

$$\begin{aligned} \log \theta_{g1} &= \text{home} + \text{att}_{h(g)} + \text{def}_{a(g)} \\ \log \theta_{g2} &= \text{att}_{a(g)} + \text{def}_{h(g)} \end{aligned}$$

The parameter *home* represents the previous observed advantage for the team hosting the match; we can assume that this effect is constant for all teams and throughout the season. However, the intensity of the score is jointly determined by the attacking and defensive capabilities of the two teams involved, represented by the parameters *att* and *def*, respectively. The indexes *h(g)* and *a(g)* identify the team playing home or away in the g-th game of the season.

It is therefore necessary to specify some suitable prior distributions for all the random parameters of the model. We can assume a flat prior distribution for the *home* parameter (described in terms of mean and precision).

$$\text{home} \sim \text{Normal}(0, 0.0001)$$

The team-specific effects are modelled as exchangeable from a common distribution.

$$\begin{aligned} \text{att}_t &\sim \text{Normal}(\mu_{\text{att}}, \tau_{\text{att}}) \\ \text{def}_t &\sim \text{Normal}(\mu_{\text{def}}, \tau_{\text{def}}) \end{aligned}$$

We need to impose the zero-sum constraint to these parameters: $\sum_{t=1}^{T} \text{att}_t = 0$ and $\sum_{t=1}^{T} \text{def}_t = 0$. In this way, the overall effects of attack and defence will be zero.

Finally, the hyper-priors of attack and defence effects are modelled independently using flat prior distributions once again:

$$\begin{aligned} \mu_{\text{att}} &\sim \text{Normal}(0, 0.001) \\ \tau_{\text{att}} &\sim \text{Gamma}(0.1, 0.1) \\ \mu_{\text{def}} &\sim \text{Normal}(0, 0.001) \\ \tau_{\text{def}} &\sim \text{Gamma}(0.1, 0.1) \end{aligned}$$

The model implemented in JAGS is as follows:

```
model {
  # Likelihoods
  for (g in 1:Ngames) {
    Y1[g] ~ dpois(theta[g,1])
    Y2[g] ~ dpois(theta[g,2])

    # Log-linear model
    log(theta[g,1]) <- home + att[hometeam[g]] + def[awayteam[g]]
    log(theta[g,2]) <- att[awayteam[g]] + def[hometeam[g]]
    }

  # Home parameter
  home ~ dnorm(0, 0.0001)

  # Team-specific effects
  for (t in 1:Nteams) {
    att.star[t] ~ dnorm(mu.att, tau.att)
    def.star[t] ~ dnorm(mu.def, tau.def)
    # Zero-sum constraints
    att[t] <- att.star[t] - mean(att.star[])
    def[t] <- def.star[t] - mean(def.star[])
  }

  # Hyper parameters priors
  mu.att ~ dnorm(0, 0.0001)
  mu.def ~ dnorm(0, 0.0001)
  tau.att ~ dgamma(0.01, 0.01)
  tau.def ~ dgamma(0.01, 0.01)
}
```

3.1 RJags

We can perform our Bayesian analysis using the *JAGS* software and interacting with it via the R package *R2Jags*. The software requires data in list format, a vector of parameters of interest and the model stored in an external file.

```
# Data for Jags
games.data <- list(hometeam = games$h.g,
                   awayteam = games$a.g,
                   Y1 = games$Yg1,
                   Y2 = games$Yg2,
                   Ngames = length(games[,1]),
                   Nteams = 20)

# Vector of parameters
param <- c('home', 'att', 'def')
```

We decided to consider the 'home' parameter and the effects of the attack and defence of all 20 Serie A teams. In the detailed analysis, for reasons of time and space, we will only focus on the parameters related to the 4 best teams of the 2021-2022 Serie A: Milan, Inter, Napoli and Juventus.

```
set.seed(1999) # seed for reproducibility

# Run the model with jags
games.jags <- jags(data = games.data,
                   parameters.to.save = param,
                   model.file = 'model_1.txt',
                   n.iter = 2000, n.chains = 3)
```

```
## module glm loaded

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 760
##    Unobserved stochastic nodes: 45
##    Total graph size: 3134
##
## Initializing model
```

We can visualize the results in a nice way, to better understand the behaviour of different clubs.

Attacking effects of Serie A teams 2021-2022

Teams	mean	2.5%	50%	97.5%
Atalanta	0.193	-0.046	0.193	0.430
Bologna	-0.161	-0.439	-0.157	0.109
Cagliari	-0.370	-0.673	-0.362	-0.093
Empoli	-0.040	-0.302	-0.037	0.201
Fiorentina	0.105	-0.141	0.108	0.347
Genoa	-0.556	-0.883	-0.553	-0.236

Hellas Verona	0.203	-0.045	0.203	0.442
Inter	0.434	0.214	0.434	0.651
Juventus	0.065	-0.193	0.068	0.313
Lazio	0.367	0.140	0.367	0.599
Milan	0.236	0.009	0.240	0.457
Napoli	0.310	0.089	0.309	0.531
Roma	0.099	-0.164	0.101	0.339
Salernitana	-0.387	-0.695	-0.382	-0.101
Sampdoria	-0.115	-0.381	-0.112	0.147
Sassuolo	0.189	-0.059	0.191	0.426
Spezia	-0.208	-0.498	-0.202	0.064
Torino	-0.132	-0.393	-0.128	0.115
Udinese	0.144	-0.101	0.145	0.374
Venezia	-0.376	-0.694	-0.375	-0.089

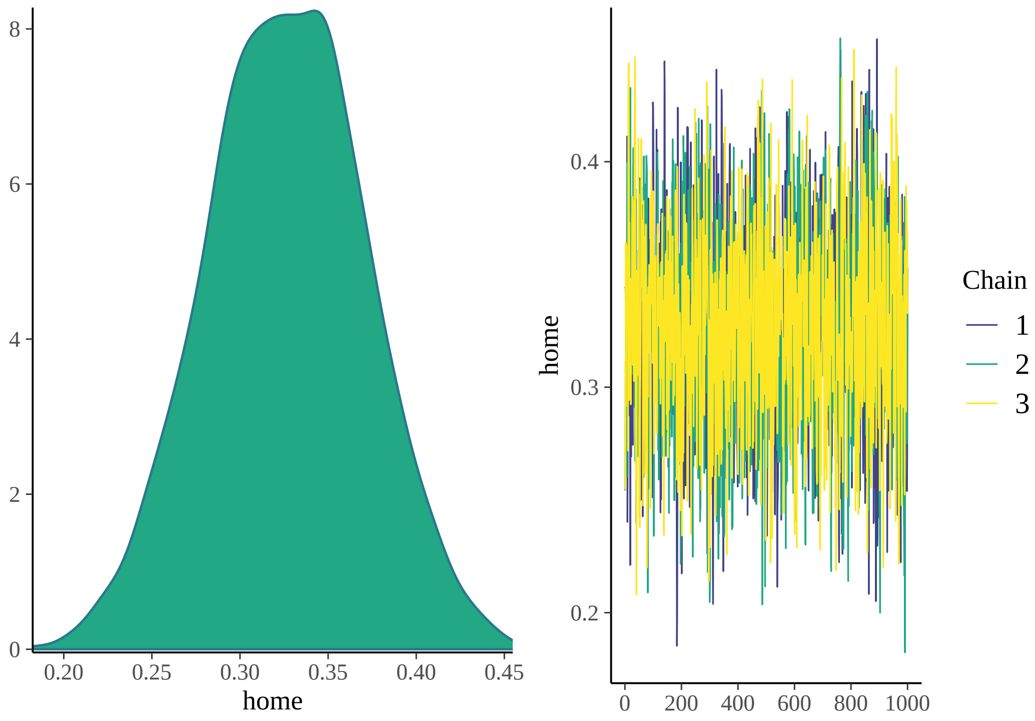
Defence effects of Serie A teams 2021-2022

Teams	mean	2.5%	50%	97.5%
Atalanta	-0.077	-0.336	-0.079	0.178
Bologna	0.022	-0.223	0.025	0.254
Cagliari	0.210	-0.025	0.210	0.441
Empoli	0.246	0.007	0.246	0.469
Fiorentina	-0.035	-0.283	-0.032	0.207
Genoa	0.092	-0.151	0.094	0.320
Hellas Verona	0.103	-0.135	0.107	0.331
Inter	-0.383	-0.704	-0.373	-0.096
Juventus	-0.289	-0.580	-0.285	-0.012
Lazio	0.092	-0.145	0.094	0.334
Milan	-0.416	-0.737	-0.410	-0.131
Napoli	-0.415	-0.742	-0.410	-0.131
Roma	-0.176	-0.437	-0.171	0.074
Salernitana	0.334	0.112	0.336	0.550
Sampdoria	0.148	-0.089	0.152	0.380
Sassuolo	0.202	-0.036	0.204	0.431
Spezia	0.256	0.018	0.255	0.488
Torino	-0.218	-0.503	-0.214	0.042
Udinese	0.082	-0.165	0.082	0.318
Venezia	0.223	0.002	0.227	0.436

From these tables, we can draw some interesting considerations. Looking at the *attacking effects*, we noticed that Inter, Lazio, Napoli and Milan are the clubs with the greatest propensity to attack and score goals. This is true, these clubs scored the most goals during the season. While clubs like Genoa, Salernitana, Venezia and Cagliari have the least propensity to attack. This makes sense, because these were the last 4 teams in the final standings of the season.

If we look at the *defence effects*, we can draw similar conclusions. Salernitana, Spezia, Empoli and Venezia are the teams with the highest propensity to concede goals, so they are not strong defences; while Milan, Napoli, Inter and Juventus have the lowest values. This confirms the tradition that in Italy it is important to have very good defences, in fact these were the top 4 teams in the final standings.

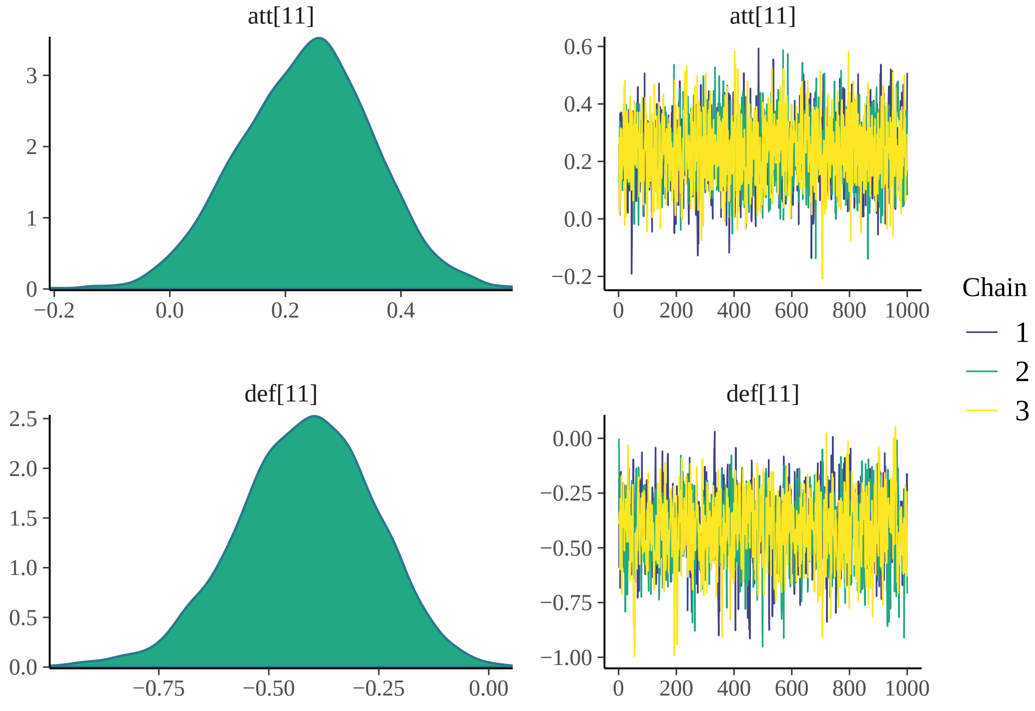
It is interesting to have a look at the *home* parameter.

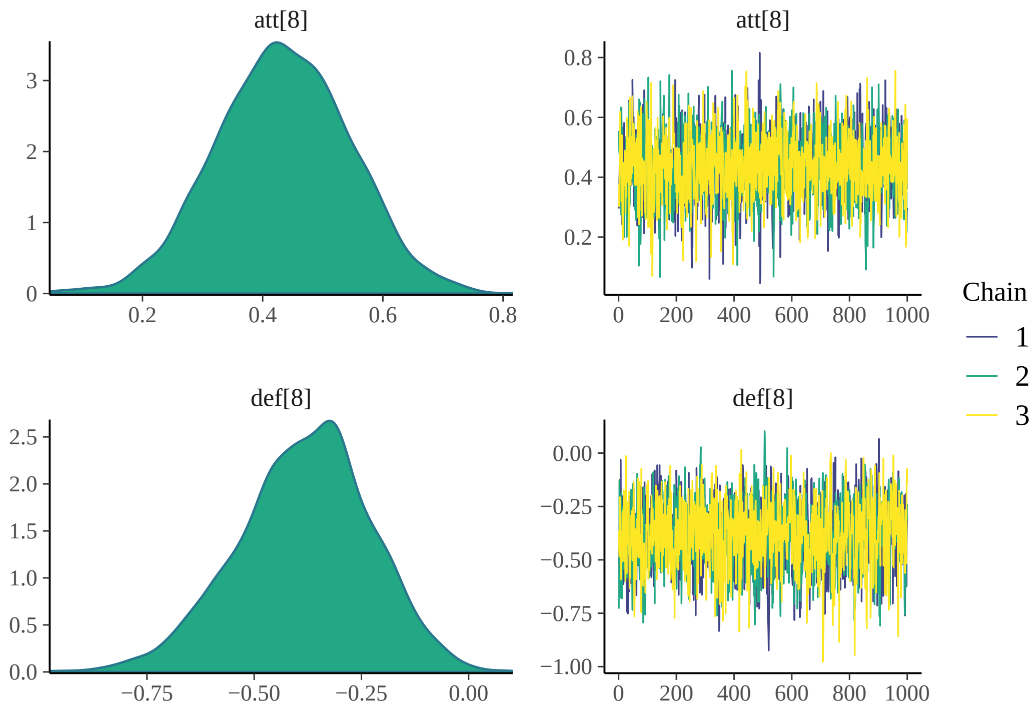


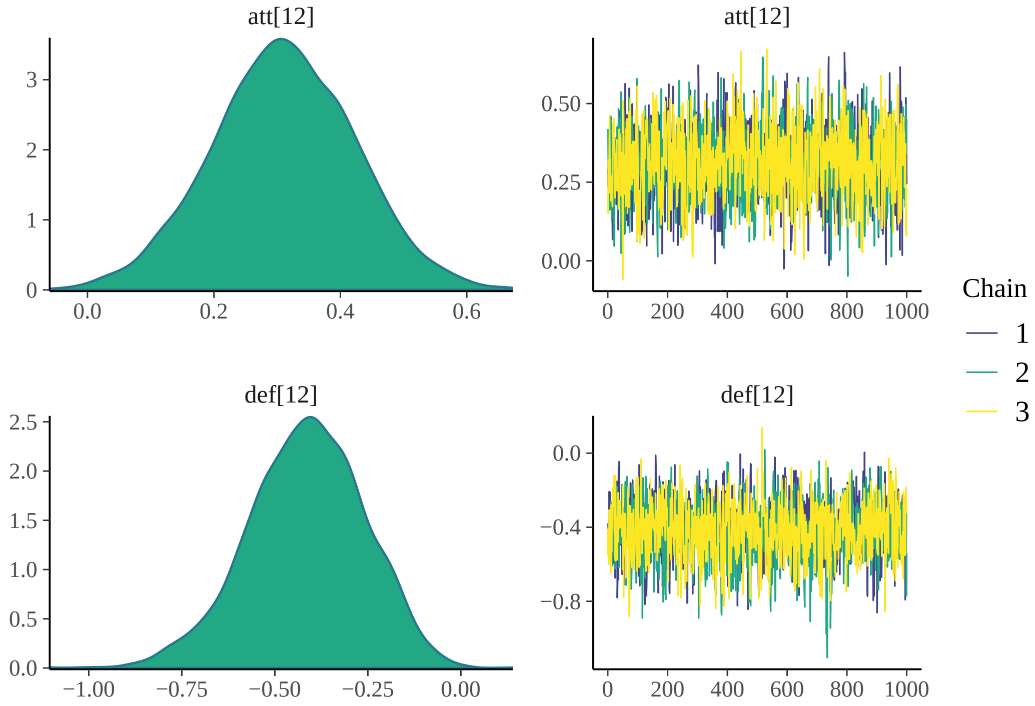
It can be seen that the home effect is clearly positive and its mean is 0.326. This means that in the 2021-2022 Serie A, the home team had a small advantage over its opponent.

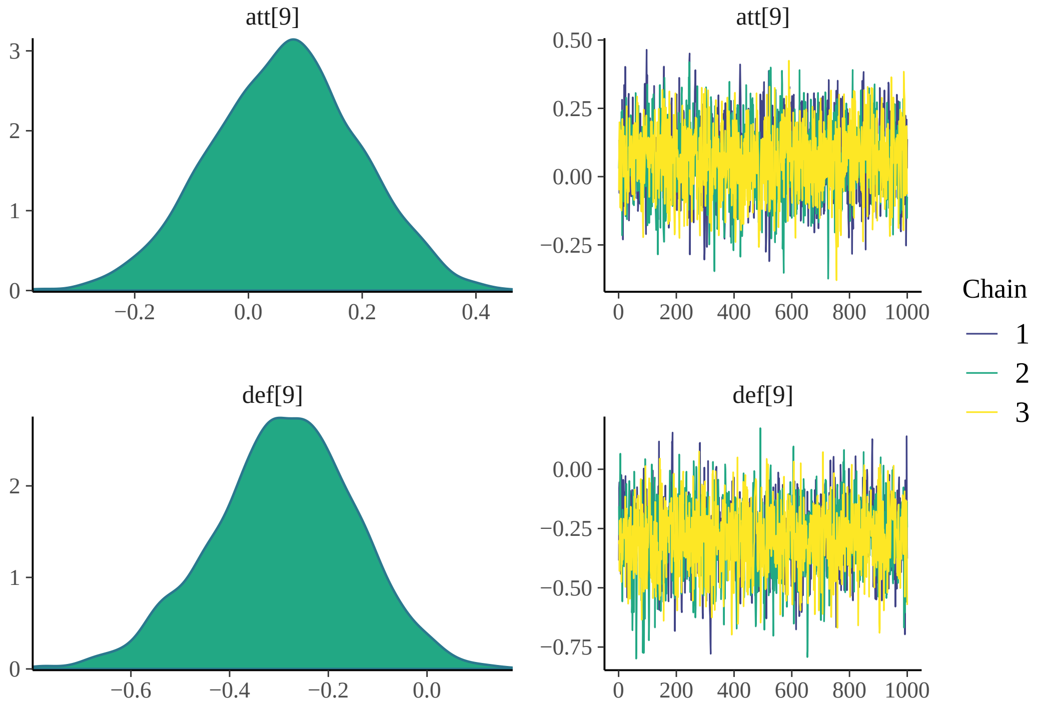
From now on, we will only consider the parameters of the top 4 clubs.

First, we can look at some basic visualizations, such as the density plots and traceplots.

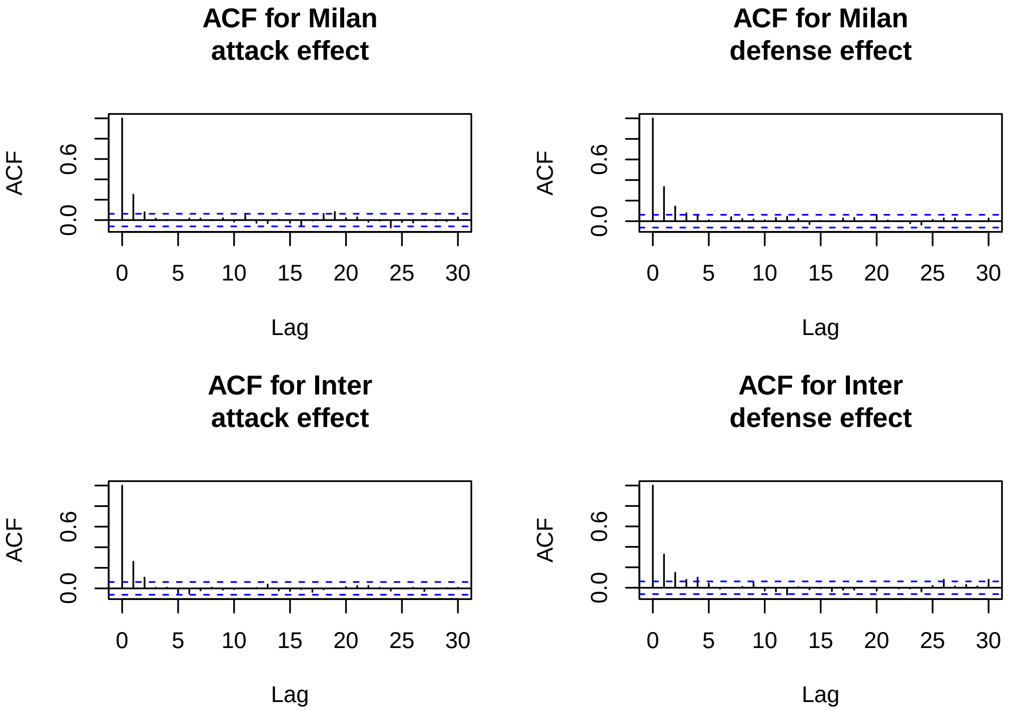


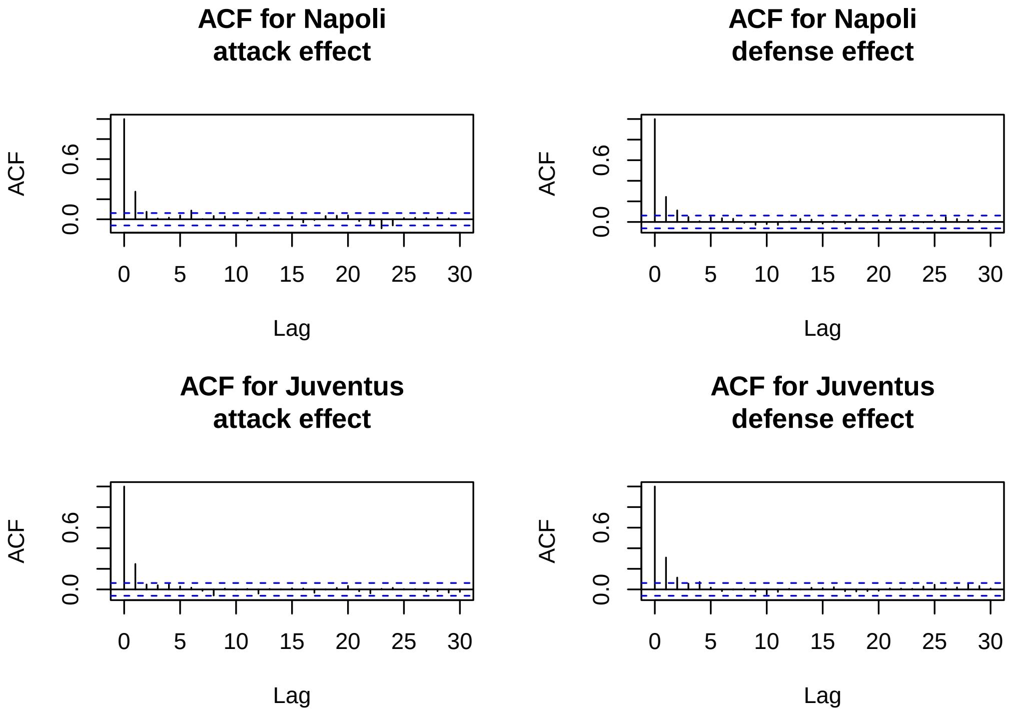

att[8]
3
2
1
0
0.2
0.4
0.6
0.8
att[8]
0.8
0.6
0.4
0.2
0
200
400
600
800
1000
Chain
1
2
3
def[8]
2.5
2.0
1.5
1.0
0.5
0.0
−0.75
−0.50
−0.25
0.00
def[8]
0.00
−0.25
−0.50
−0.75
−1.00
0
200
400
600
800
1000


att[12]
3
2
1
0
0.0
0.2
0.4
0.6
att[12]
0.50
0.25
0.00
0
200
400
600
800
1000
Chain
1
2
3
def[12]
2.5
2.0
1.5
1.0
0.5
0.0
−1.00
−0.75
−0.50
−0.25
0.00
def[12]
0.0
−0.4
−0.8
0
200
400
600
800
1000



As we can see, the results seem to be acceptable, the history of the chains is what we should see, fluctuations and apparently no correlation between the samples. Furthermore, all eight parameters have a normal distribution, as expected. However, let us take a closer look at the autocorrelation of the process:





All these autocorrelations goes to zero with an acceptable lag. In general, one notices slightly worse behaviour from defence effects, which require more lag to keep the autocorrelation low.

3.1.1 Approximation Error

We now want to calculate the approximation error of the parameters as the square root of:

$$\mathbb{V}[\hat{I}_t] = \frac{\mathbb{V}_\pi[h(\theta_i)]}{t_{\text{eff}}}$$

where $\mathbb{V}_\pi[h(\theta_i)]$ is the variance of the parameter values during the chain and t_{eff} is the effective sample size (ESS). It is used to have a sort of exchange rate between dependent and independent samples, to show how much independent sample a given MCMC can correspond to. We can calculate it using the *effectiveSize* function of the *coda* package.

```r
# Initialize the approximation error matrix
app.err <- matrix(nrow = 4, ncol = 2)
for (i in 1:4) {
  # Attack effect
  app.err[i,1] = round(sqrt(var(att.list[,team_codes[i]])/effectiveSize(att.list[,team_codes[i]])), 5)
  # Defense effect
  app.err[i,2] = round(sqrt(var(def.list[,team_codes[i]])/effectiveSize(def.list[,team_codes[i]])), 5)
}
```

Approximation error

Teams	Attack	Defense
Milan	0.00210	0.00267
Inter	0.00203	0.00282

Napoli	0.00212	0.00285
Juventus	0.00237	0.00266

As expected, the higher the approximation error, the slower the decrease of the ACF, in fact, it can be observed that defence effects always have higher values than attack effects. In particular, the uncertainty of Napoli's defence effect is the highest.

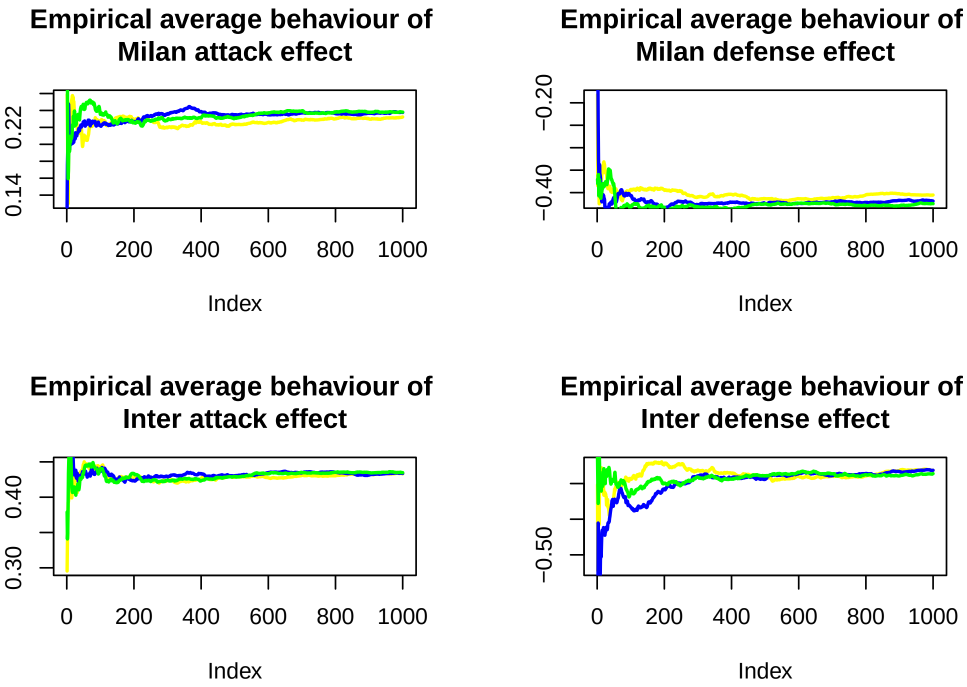
3.1.2 Empirical Mean Behaviour

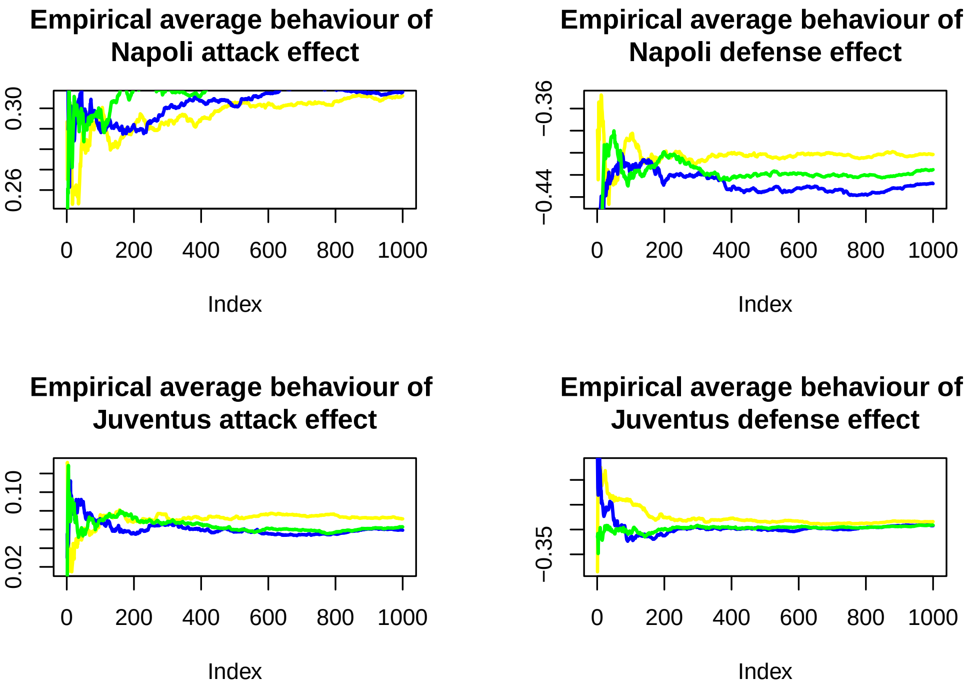
We can evaluate the behaviour of the empirical average by observing the evolution of $\hat{I}_t$ for each chosen parameter and for each chain. The behaviour of the empirical averages is evaluted as follows, with increasing $t = 1, \ldots, T$ and T is the total number of simulations.

$$\hat{I}_t = \frac{1}{t} \sum_{i=1}^{t} h(\theta_i)$$

where θ_i represents the chosen parameter.

The idea is very simple, if the chain has almost reached stationarity, we should not see large fluctuactions around the final real value.





Graphically, we can assess that the process from a certain point onwards will be stationary and that the chains agree on the final result almost all the time. In particular, the chains of the Napoli defence effect appear not to converge to the same result at each chain. This result reflects the value of uncertainty assessed in the previous analysis.

3.1.3 Posterior Uncertainty

We measure uncertainty using the variability of the parameter with respect to its absolute expectation.

$$U_i = \frac{\mathbb{V}[\hat{\theta}_i]^{1/2}}{|\mathbb{E}[\hat{\theta}_i]|}$$

Of course, we can use the plug-in estimators (empirical sd and expectation) provided by the chains:

```r
# Posterior uncertainty
post.unc <- matrix(nrow = 4, ncol = 2)
for (i in 1:4) {
  # Expected values
  mu.sim.att = mean(att.list[,team_codes[i]])
  mu.sim.def = mean(def.list[,team_codes[i]])
  # Attack effect
  post.unc[i,1] = round(sqrt(var(att.list[,team_codes[i]])) / abs(mu.sim.att), 5)
  # Defense effect
  post.unc[i,2] = round(sqrt(var(def.list[,team_codes[i]])) / abs(mu.sim.def), 5)
}
```

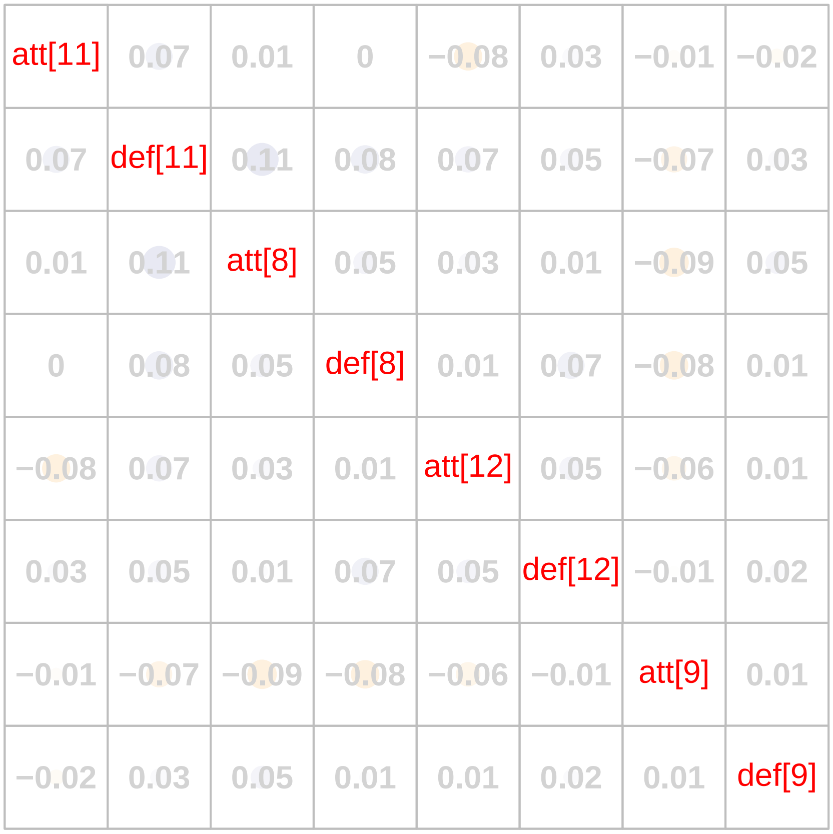
Posterior uncertainty

Teams	Attack	Defense
Milan	0.48768	0.37590
Inter	0.25641	0.40316
Napoli	0.36178	0.37567
Juventus	2.00936	0.50472

The highest posterior uncertainty is registered by the effect of Juventus' attack.

3.1.4 Parameters Correlation

To see the correlation between the chosen parameters, we can calculate the empirical covariance matrix using the chains and take the *corrplot*. In this case, we are looking for a linear correlation.



att[11]	0.07	0.01	0	−0.08	0.03	−0.01	−0.02
0.07	def[11]	0.11	0.08	0.07	0.05	−0.07	0.03
0.01	0.11	att[8]	0.05	0.03	0.01	−0.09	0.05
0	0.08	0.05	def[8]	0.01	0.07	−0.08	0.01
−0.08	0.07	0.03	0.01	att[12]	0.05	−0.06	0.01
0.03	0.05	0.01	0.07	0.05	def[12]	−0.01	0.02
−0.01	−0.07	−0.09	−0.08	−0.06	−0.01	att[9]	0.01
−0.02	0.03	0.05	0.01	0.01	0.02	0.01	def[9]

We observe that the team-specific effects are not correlated at all; we can expect this result because these effects are not correlated with each other but only relate to team performance.

3.2 Predictions

We can compare the data of the real 2021-2022 Serie A with a simulated Serie A based on the values of the obtained model. Therefore, we decided to simulate each match of the competition taking into account all previously evaluated effects to predict the goals scored by each team during a given match.

Based on the simulated result of each match, we decided to award a win if one team scores 1 goal more than the opponent, otherwise it will be a draw. As for the real competition, for each match we awarded 3 points to the winner, 1 point to both teams in case of a draw and 0 points to the loser.

Finally, we are able to calculate a final standings with total points collected, total goals scored and conceded.

Simulated Serie A 2021-2022

Teams	Points	Goal Scored	Goal Conceded
Inter	82	73	31
Napoli	72	65	30
Milan	62	60	31
Lazio	54	67	50
Juventus	52	50	35
Atalanta	51	57	43
Roma	51	52	39
Fiorentina	48	52	45
Hellas Verona	47	57	52
Udinese	47	54	51
Sassuolo	44	56	57
Torino	39	41	38
Bologna	34	40	48
Sampdoria	31	41	55
Empoli	27	44	60
Genoa	26	27	53
Cagliari	25	32	59
Spezia	25	37	61
Venezia	25	32	60
Salernitana	23	31	67

Real Serie A 2021-2022

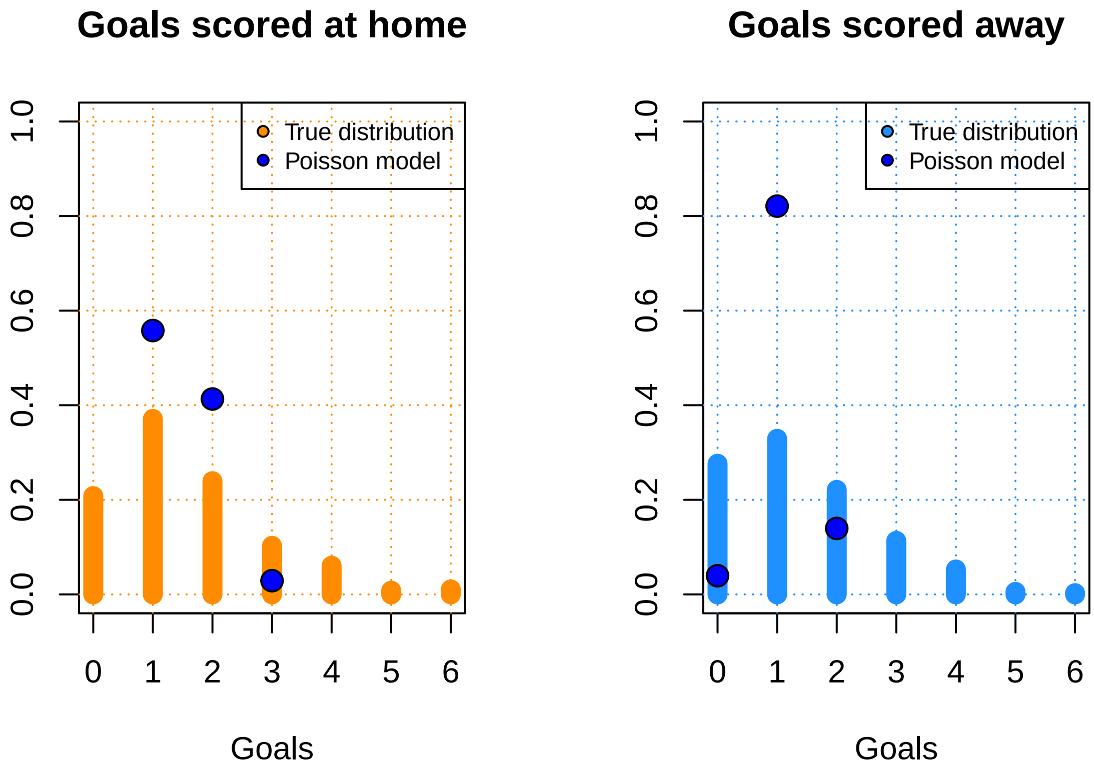
Teams	Points	Goal Scored	Goal Conceded
Milan	86	69	31
Inter	84	84	32
Napoli	79	74	31
Juventus	70	57	37
Lazio	64	77	58
Roma	63	59	43
Fiorentina	62	59	51
Atalanta	59	65	48
Hellas Verona	53	65	59
Torino	50	46	41
Sassuolo	50	64	66
Udinese	47	61	58
Bologna	46	44	55
Empoli	41	50	70
Sampdoria	36	46	63
Spezia	36	41	71
Salernitana	31	33	78
Cagliari	30	34	68
Genoa	28	27	60
Venezia	27	34	69

There are definitely differences between the real and simulated final standings, starting with the winner. In the simulated Serie A, Inter won, while in the real one Milan won. However, the overall league situation is respected and the differences between the top and bottom teams are obvious.

This model is based only on the goals scored by the teams, which is why the final ranking of the simulated Serie A follows the ranking of the best goal difference in the real one. In fact, if we order the teams according to goals scored, the two rankings will coincide.

3.3 Bayesian Inference

We can make a Bayesian inference on the goals scored during the season, using the results of our first evaluated model. We can divide between goals scored at home and goals scored away, just as we did in our model. In the end, we can compare the estimates with the actual distributions of goals scored.



It can be seen that the predictions of goals scored are more concentrated around low scores and never predict matches with a high number of goals scored by a team. Interestingly, in the model, the home team always scores at least 1 goal in the match, which can be interpreted as the effect of the previously described home parameter. As for away goals scored, the distribution is really concentrated around 1 goal and no team in any match scores more than 2 away goals. Ultimately, the advantage of the home team is also confirmed by the model, but it cannot predict the outliers of the distribution.

4. Alternative Model

As an alternative model, we decided to model the number of goals scored as Negative Binomials, but not to consider the home effect. The Negative Binomial also allows us to consider the case where the data are said to be overdispersed, i.e. when the variance exceeds the mean. Overdispersion is expected for events where the first occurrence makes a second occurrence more likely, even if still random. In this case, we need further reparametrization to obtain an appropriate form, which we performed following the article by

Derpanopoulos (2016).

$$\begin{aligned} Y_{g1} &\sim \text{NegBin}(p_{g1}, r) \\ Y_{g2} &\sim \text{NegBin}(p_{g2}, r) \end{aligned}$$

The NB densities are parametrized for the g-th match with p_g and r. The parameters p_{g1} and p_{g2} are the success parameters and for each match are defined as:

$$\begin{aligned} p_{g1} &= \frac{r}{r + \lambda_{g1}} \\ p_{g2} &= \frac{r}{r + \lambda_{g2}} \end{aligned}$$

Thus, λ_{g1} and λ_{g2} are modeled as before, assuming a log-linear random effects model. In this case, we have an additional parameter r which is the size parameter, the target for number of successful trials. It is assumed to be uniform over a wide interval.

$$r \sim \text{Unif}(0, 50)$$

All other parameters, hyper-parameters and the prior distribution remain unchanged. The model implemented in JAGS is as follows:

```
model{
    # Likelihoods
    for (g in 1:Ngames) {
      Y1[g] ~ dnegbin(p[g,1],r)
      Y2[g] ~ dnegbin(p[g,2],r)

      # Success parameters
      p[g,1] <- r/(r+lambda[g,1])
      p[g,2] <- r/(r+lambda[g,2])

      # Log-linear model
      log(lambda[g,1]) <- att[hometeam[g]] + def[awayteam[g]]
      log(lambda[g,2]) <- att[awayteam[g]] + def[hometeam[g]]
    }

  # Size parameter
  r ~ dunif(0, 50)

  # Team-specific effects
  for (t in 1:Nteams) {
    att.star[t] ~ dnorm(mu.att, tau.att)
    def.star[t] ~ dnorm(mu.def, tau.def)
    # Zero-sum constraints
    att[t] <- att.star[t] - mean(att.star[])
    def[t] <- def.star[t] - mean(def.star[])
  }

  # Hyper parameters priors
  mu.att ~ dnorm(0, 0.0001)
  mu.def ~ dnorm(0, 0.0001)
  tau.att ~ dgamma(0.01, 0.01)
  tau.def ~ dgamma(0.01, 0.01)
}
```

4.1 RJags

Using the same data and parameter vector as before, we can run our new model with *R2Jags*.

```r
set.seed(1999) # seed for reproducibility

# Parameter vector
param2 <- c('att', 'def')

# Run the model with jags
games.jags2 <- jags(data = games.data,
                    parameters.to.save = param2,
                    model.file = 'model_2.txt',
                    n.iter = 2000, n.chains = 3)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 760
##    Unobserved stochastic nodes: 45
##    Total graph size: 3135
##
## Initializing model
```

We can visualize the results of team-specific effects with two tables.

Attack effects of Serie A teams 2021-2022 (NegBin)

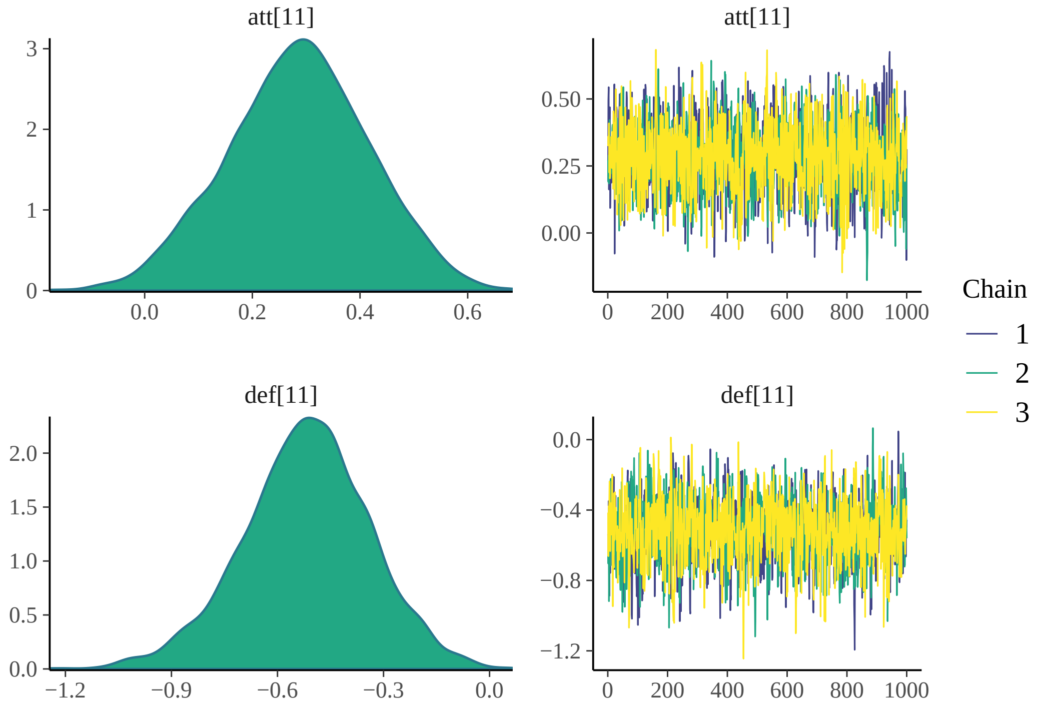
Teams	mean	2.5%	50%	97.5%
Atalanta	0.229	-0.033	0.230	0.482
Bologna	-0.186	-0.516	-0.179	0.106
Cagliari	-0.443	-0.790	-0.441	-0.111
Empoli	-0.035	-0.328	-0.031	0.238
Fiorentina	0.131	-0.150	0.131	0.404
Genoa	-0.683	-1.074	-0.682	-0.325
Hellas Verona	0.246	-0.006	0.250	0.482
Inter	0.505	0.273	0.505	0.726
Juventus	0.080	-0.191	0.085	0.342
Lazio	0.429	0.187	0.431	0.670
Milan	0.285	0.025	0.288	0.536
Napoli	0.363	0.113	0.366	0.608
Roma	0.122	-0.158	0.123	0.397
Salernitana	-0.475	-0.846	-0.471	-0.146
Sampdoria	-0.130	-0.437	-0.127	0.154
Sassuolo	0.230	-0.044	0.232	0.479
Spezia	-0.248	-0.565	-0.245	0.051
Torino	-0.151	-0.454	-0.145	0.134
Udinese	0.172	-0.089	0.174	0.419
Venezia	-0.441	-0.791	-0.439	-0.113

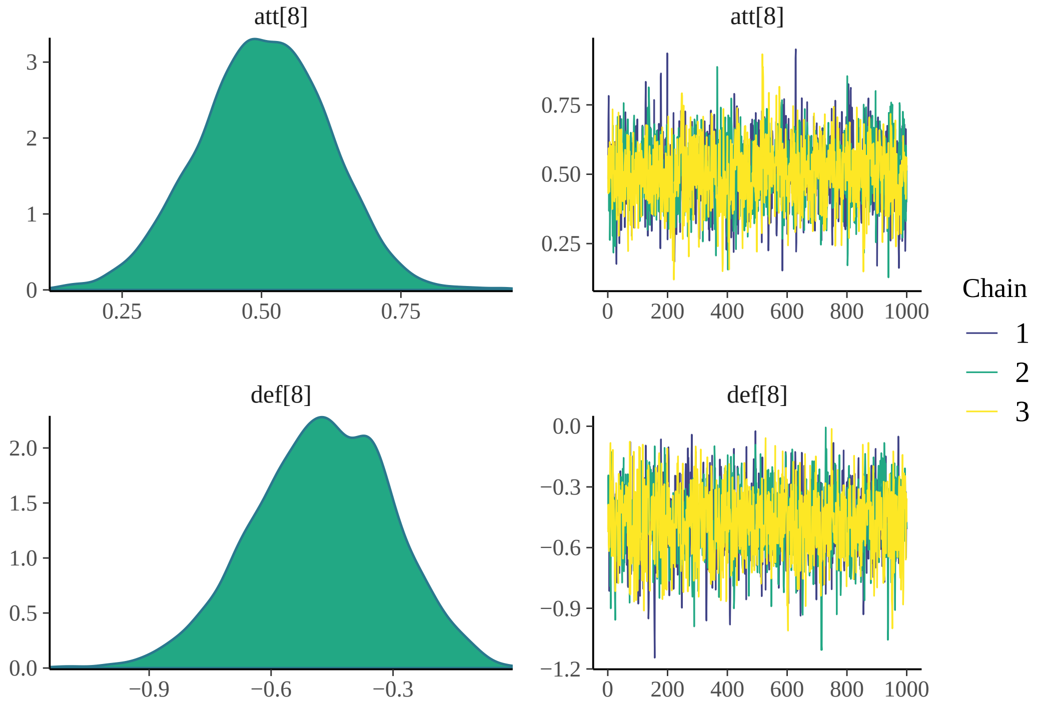
Defense effects of Serie A teams 2021-2022 (NegBin)

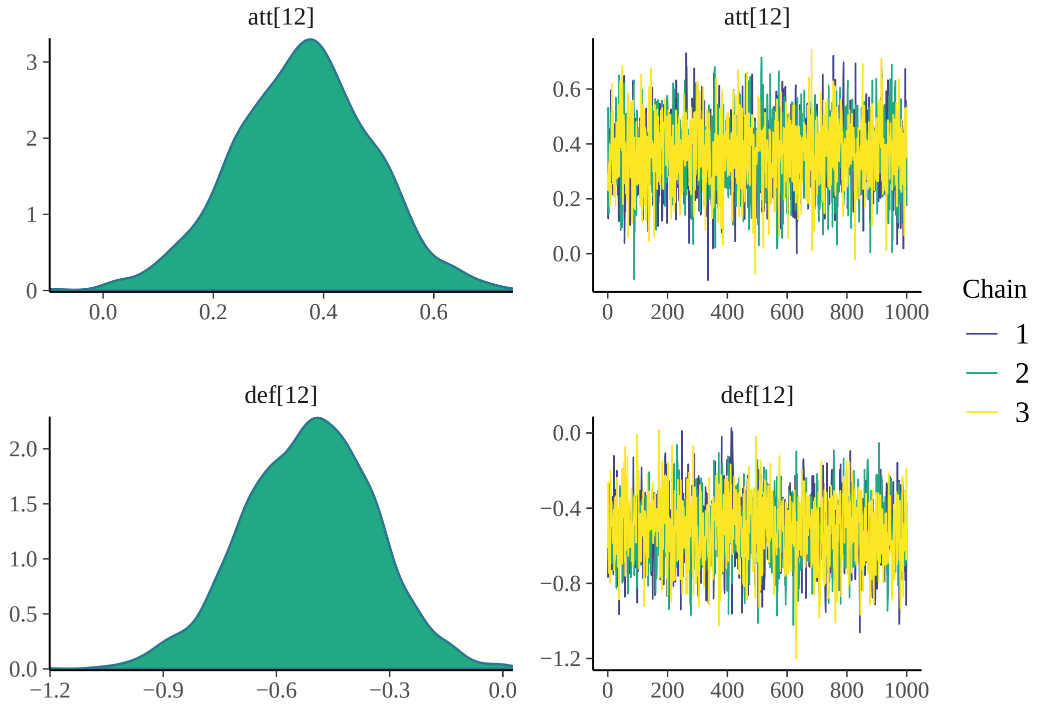
Teams	mean	2.5%	50%	97.5%
Atalanta	-0.091	-0.395	-0.087	0.192
Bologna	0.040	-0.246	0.042	0.297
Cagliari	0.254	0.012	0.256	0.493
Empoli	0.298	0.057	0.301	0.538
Fiorentina	-0.030	-0.326	-0.024	0.245
Genoa	0.110	-0.164	0.113	0.369
Hellas Verona	0.132	-0.122	0.132	0.389
Inter	-0.474	-0.816	-0.469	-0.157
Juventus	-0.360	-0.692	-0.357	-0.046
Lazio	0.127	-0.159	0.128	0.389
Milan	-0.519	-0.891	-0.513	-0.177
Napoli	-0.514	-0.867	-0.507	-0.183
Roma	-0.210	-0.508	-0.208	0.070
Salernitana	0.400	0.168	0.401	0.637
Sampdoria	0.179	-0.082	0.182	0.422
Sassuolo	0.244	-0.012	0.249	0.483
Spezia	0.306	0.063	0.311	0.539
Torino	-0.261	-0.574	-0.260	0.031
Udinese	0.100	-0.178	0.103	0.370
Venezia	0.267	0.011	0.268	0.514

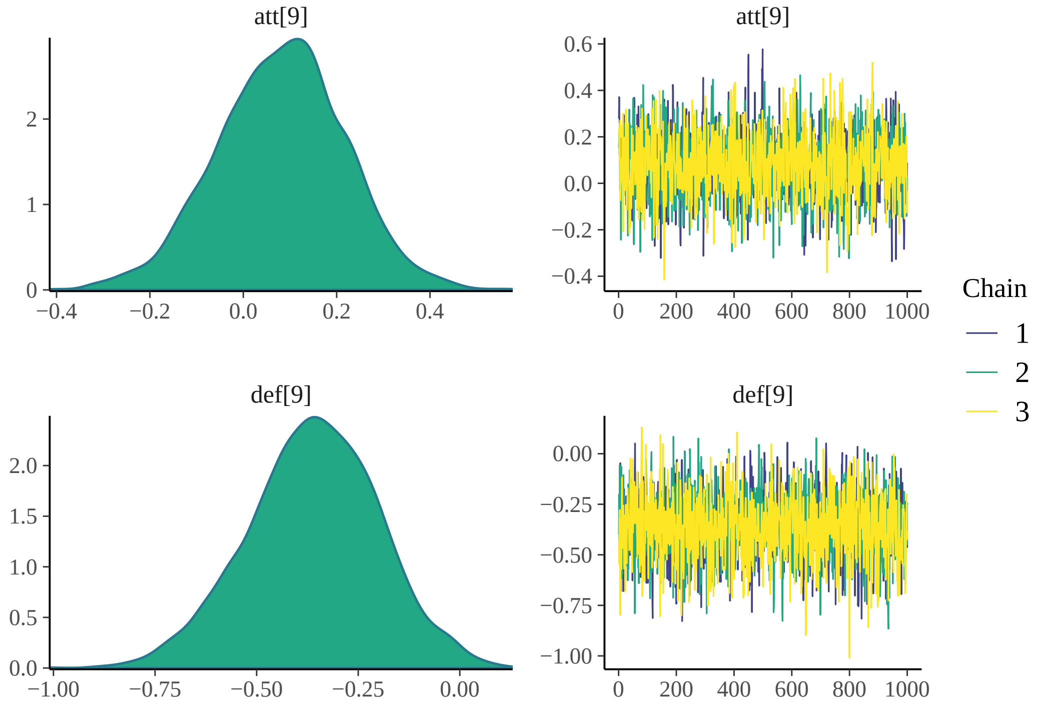
Regarding the attack effects, we noticed that we have more or less the same ranking as the previous model, but the effects are strengthened for all teams. Also for defence effects we have in general stronger values. The positive effects are now higher and the negative effects lower. These increased values are probably due to the lack of the home effect.

Let us look some basic visualization such as density plots and traceplots for this new model, focusing only on the effects of the 4 top Serie A teams: Milan, Inter, Napoli and Juventus.

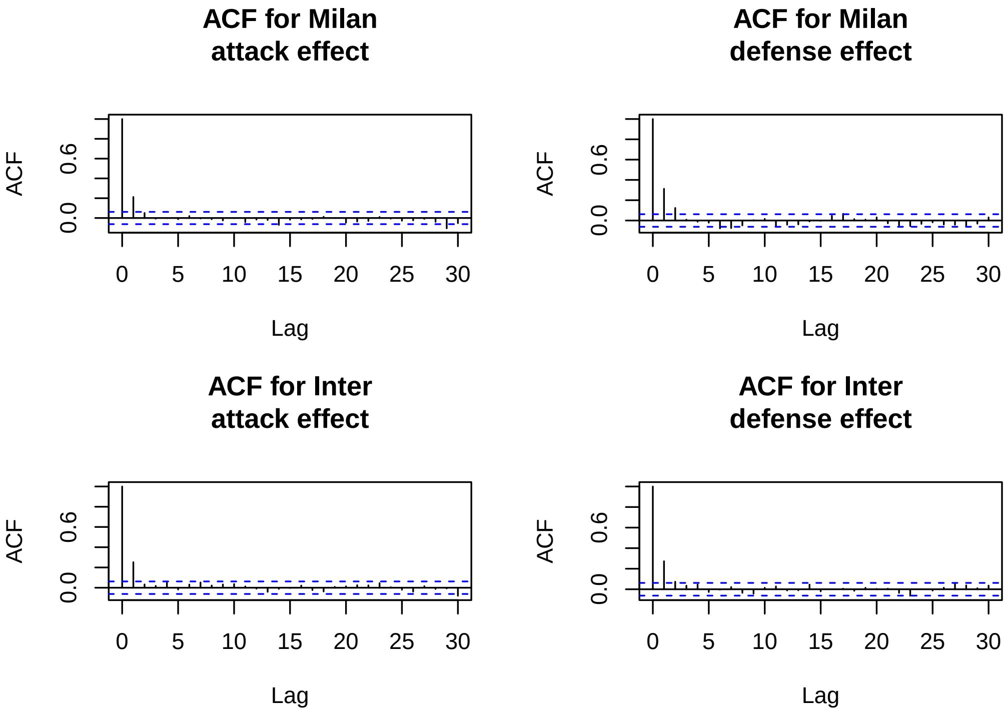

att[11]
3
2
1
0
0.0
0.2
0.4
0.6
att[11]
0.50
0.25
0.00
0
200
400
600
800
1000
Chain
1
2
3
def[11]
2.0
1.5
1.0
0.5
0.0
−1.2
−0.9
−0.6
−0.3
0.0
def[11]
0.0
−0.4
−0.8
−1.2
0
200
400
600
800
1000

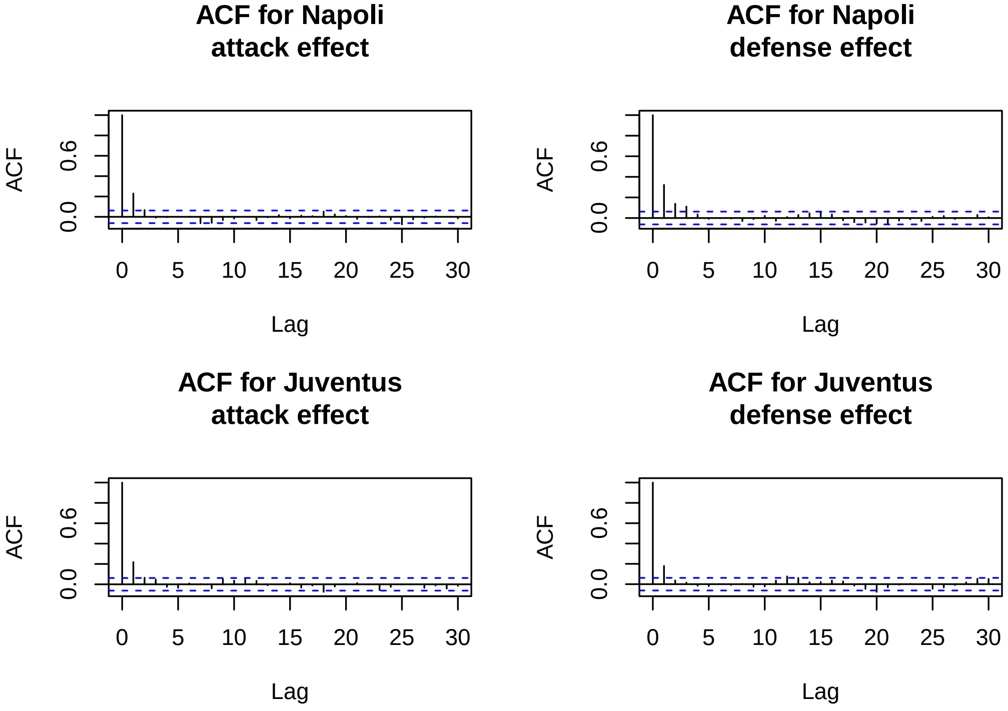

att[8]
3
2
1
0
0.25
0.50
0.75
att[8]
0.75
0.50
0.25
0
200
400
600
800
1000
Chain
1
2
3
def[8]
2.0
1.5
1.0
0.5
0.0
−0.9
−0.6
−0.3
def[8]
0.0
−0.3
−0.6
−0.9
−1.2
0
200
400
600
800
1000





Again, the results seem to be acceptable, we apparently have no correlation between the samples and there are the expected fluctuactions in the history of the chains. Let us take a look at the autocorrelations of the process:





All these autocorrelations goes to zero with an acceptable lag. In general, the slightly worse behaviour of defence effects, which require more lag to keep the autocorrelation low, is confirmed.

4.2 Comparison methods

To compare our two models, we can use two Penalized Likelihood Criteria: Deviance Information Criterion (DIC) and Akaike Information Criterion (AIC). Both are based on deviance, which can be calculated as $D(y, \theta) = -2 \log f(y|\theta)$, but is often evaluated at a "representative point" such as the posterior mean or mode: $D_{\hat{\theta}}(y) = D(y, \hat{\theta}(y))$.

Let us now compare the two models using the *DIC*. The idea is that models with smaller DIC are preferred to those with higher DIC. The Deviance Information Criterion is calculated as follows:

$$\text{DIC} = D_{\hat{\theta}}(y) + 2p_D$$

where p_D represents the effective number of parameters used and serves to penalize model complexity and $D_{\hat{\theta}}(y)$ is the deviance of the model. Obviously, DIC can only be used as a comparative index between models evaluated on the same dataset, it has no absolute meaning.

Deviance Information Criterion - DIC

Models	DIC
Model 1: Poisson	2257.433
Model 2: NegBin	2302.332

The difference between the two models allows us to state that the first model is preferable, as the DIC of the first model appears to be lower. Moreover, the Poisson model contains fewer parameters than the Negative Binomial model.

We can try to make the same comparison using the *AIC*. Again, the idea is to select the model with the lowest value. This method is often used in frequentist analysis and focuses on models that have good "out-of-sample" predictive ability. It is calculated as:

$$\text{AIC} = D_{\hat{\theta}}(y) + 2p$$

where $D_{\hat{\theta}}(y)$ is the deviance of the model and p represents the dimension of the parameter space. Again, the aim is to penalize model complexity.

Akaike Information Criterion - AIC

Models	AIC
Model 1: Poisson	2296.138
Model 2: NegBin	2340.657

We get the same results as before, the Poisson model has the lowest value. One difference between the two models was the presence of the home effect: looking at these results, it seems better to include it into the model.

4.3 Alternative predictions

We can try to simulate the Serie A season with the Negative Binomial model, to see if there are any relevant differences with the previous simulated season and the real one. This is the final standings obtained:

Simulated Serie A 2021-2022 (NegBin)

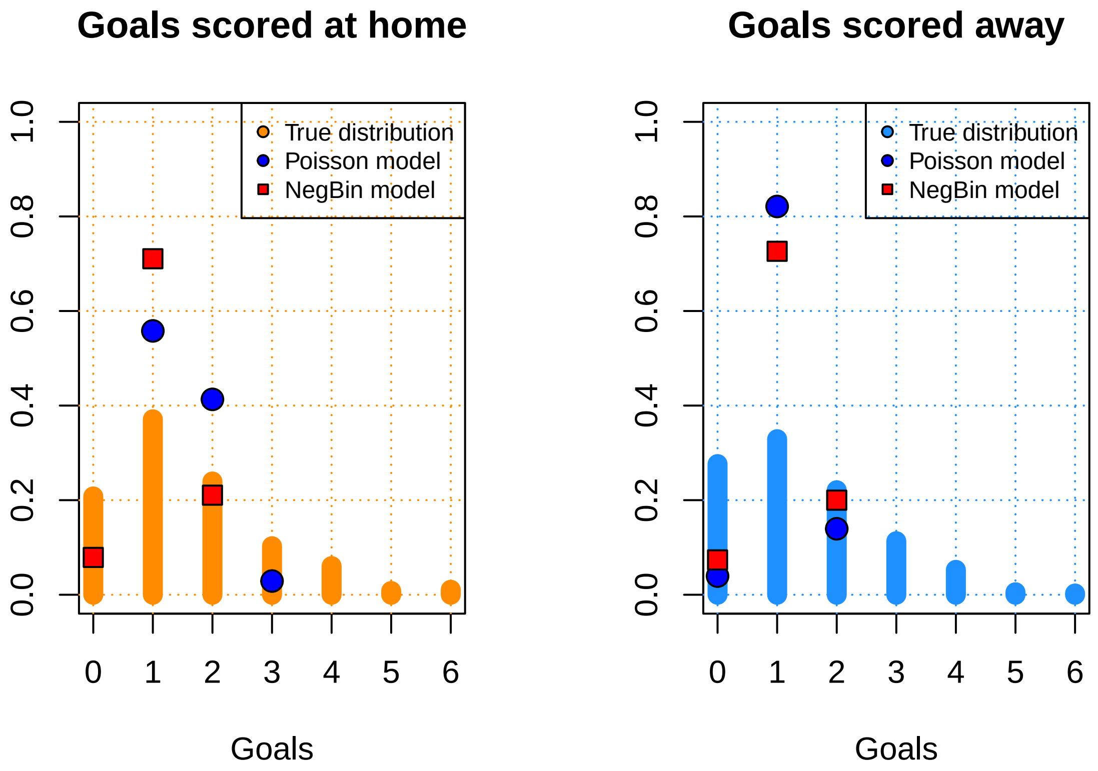
Teams	Points	Goal Scored	Goal Conceded
Inter	82	67	24

Napoli	70	58	23
Milan	66	54	23
Lazio	58	60	44
Atalanta	50	50	36
Fiorentina	42	45	39
Juventus	42	44	28
Roma	42	45	32
Hellas Verona	40	50	45
Udinese	40	47	44
Sassuolo	38	49	50
Torino	38	34	31
Bologna	36	33	42
Empoli	32	38	54
Sampdoria	32	34	48
Genoa	30	20	46
Spezia	30	30	55
Cagliari	28	25	53
Venezia	28	25	53
Salernitana	16	24	61

As can be seen, the shape of this Serie A simulation is very similar to the previous one. As expected from the two DIC measures, the two models are very close to each other. Both points and goals are almost the same as before, some differences in the points obtained are not significant in the final standings because the order remains more or less the same.

4.4 Bayesian Inference

We can make some Bayesian inference as before and compare the results with those obtained previously.



Comparing the two models, it is immediately noticeable that the Negative Binomial model has almost the same shape both at home and away, so the “natural” difference between the two distributions is missing. For away goals scored, the predictions are very close to Poisson’s. This is interesting because it gives us an idea of the real relevance of the ‘home’ effect in these models.

5. Frequentist Analysis

We can propose a frequentist alternative to our Bayesian analysis. We can consider the fact that goals scored are expressed in the form of a log-linear model with respect to some specific effects. Then, we organize in a new dataframe the home effect, the team-specific effects and the goals scored in each game of the season, in order to train a *Generalized Linear Model* to predict the number of goals scored.

In particular, we considered a *glm* with a Poisson function as the distribution family and a *log*-link function. In this way, the mean function is of the form $\mu = \exp(X\beta)$, which corresponds exactly to the specified form of our original model.

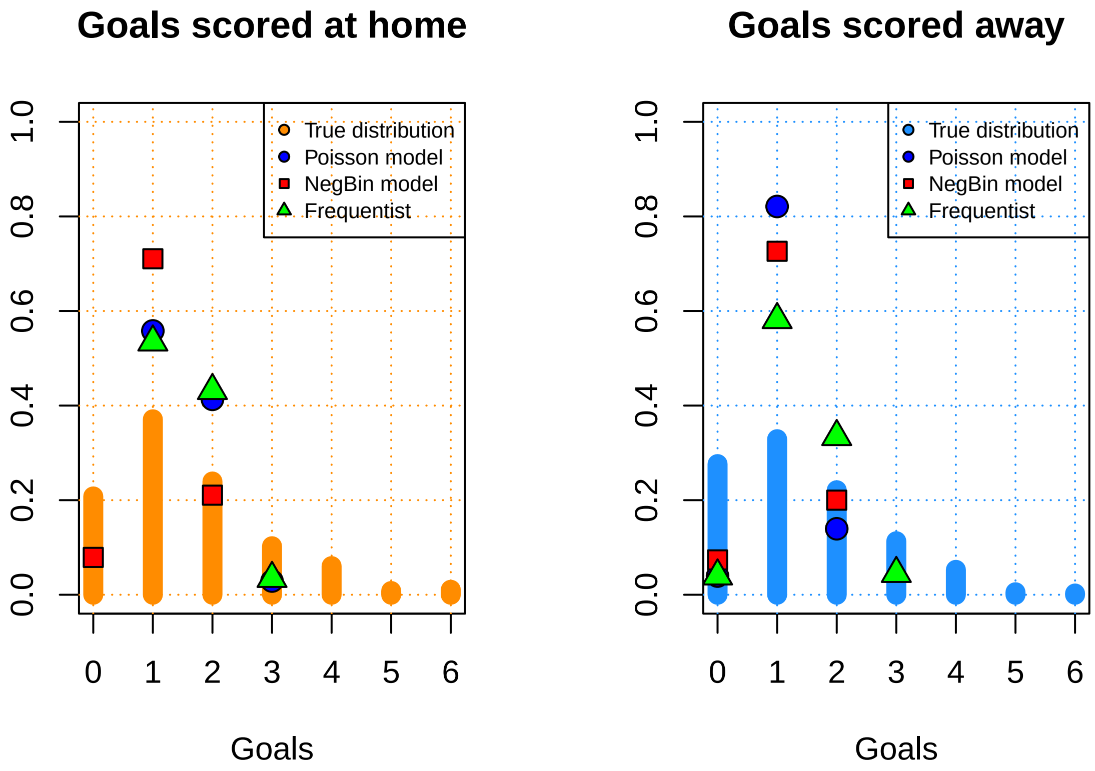
This is the model for goals scored at home:

```r
model.h <- glm(Yg1 ~ h + att.h + def.a, family=poisson(link="log"), data=df)
```

This is the model for goals scored away:

```r
model.a <- glm(Yg2 ~ att.a + def.h, family=poisson(link="log"), data=df)
```

Finally, we can compare the predictions of the fitted values of goals scored with the predictions obtained in the Bayesian framework.



We can see that for goals scored at home, the predictions are quite similar to those of Poisson model. While for away goals scored, the resulting predictions are different: they are less concentrated than the others and the range of values is wider, there are also some matches with 3 goals scored by the away team. However, goals scored at home always seem to be more numerous than those scored away. Interestingly, there is no model that can correctly predict matches in which a team scored 4, 5 or 6 goals; it probably happened too few times in the last championship. Only the Negative Binomial model predicted some matches with zero goals for the home team (and it was the only one in which the home parameter was not considered), otherwise the models always predict at least one goal for the home team, although this was by no means always true in the actual data.

To see if these differences are relevant, we can try to simulate the season with these new data as well and see what happens.

Simulated Serie A 2021-2022 (Frequentist)

Teams	Points	Goal Scored	Goal Conceded
Inter	92	85	32
Napoli	80	74	32
Milan	76	69	32
Lazio	62	77	57
Atalanta	61	65	48
Juventus	58	57	37
Roma	57	59	42
Hellas Verona	52	65	59
Fiorentina	49	58	50
Udinese	46	61	57
Torino	39	46	41

Sassuolo	38	63	66
Bologna	33	44	55
Sampdoria	28	46	63
Empoli	27	49	70
Genoa	23	28	59
Spezia	23	41	72
Cagliari	19	34	68
Venezia	19	34	69
Salernitana	15	33	79

As can be seen, the overall ranking is no different from the previous two. We noticed that the top teams tend to have more points and the bottom teams fewer points. The reason may be that there are fewer draws, so the difference between goals scored and goals conceded in a single match is generally higher. Perhaps this kind of approach overestimates the performance of a team with a good attack and defence, penalizing the performances of the other teams in the league too much.

6. Conclusions

The models presented in this project are simple applications of Bayesian hierarchical modelling. For simplicity, these predictions are based only on the number of goals scored and conceded during the entire season by each team. Of course, to obtain a more accurate model, many other variables can be included to capture the variability of the teams' form during the season (injuries, suspensions, etc.). In fact, the two predicted final standings reflect more the real “goal difference ranking” than the actual standings. In any case, the results obtained are realistic and give a clear idea of the last Serie A championship.

Appendix

```sql
-- Query used on MySQL Workbench to extract Serie A 2021-2022 data

SELECT c1.pretty_name, c2.pretty_name, g.home_club_id, g.away_club_id, g.home_club_goals,
       g.away_club_goals, g.round
FROM games AS g, clubs AS c1, clubs AS c2
WHERE g.season = 2021 AND
      g.competitions_id = 'IT1' AND
      c1.id = g.home_club_id AND
      c2.id = g.away_club_id
ORDER BY g.round;
```

References

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

Derpanopoulos G. (2016), ‘Count Models in JAGS’

Karlis, D. & Ntzoufras, I (2003), ‘Analysis of sports data by using bivariate Poisson models’, *Journal of the Royal Statistical Society* D 52, 381-393.

Pollard R., Benjamin B., Reep C. (1977), ‘Sport and the negative binomial distribution’, *Optimal strategies in sport. Eds: Ladany S.P., Machol R.E.Cleveland: North-Holland Publishing Company*, 188-195.