CHATZOPOULOS GERASIMOS

BAYESIAN STATISTICAL MODELLING

WATER POLO EURO TOURNAMENT

Table of Contents

1. WATER POLO

Water polo is a competitive, aquatic sport that has been played around the world for over 100 years. Being the first team sport in the Olympic Games, water polo has been a staple in high-level competitive sports at all the competitive levels. The landscape of sports statistics has changed considerably over the past twenty years. Real-time sports performance analysis is a crucial aspect of matches in major sports around the world. Professional sports have seen massive improvements in the way player performance, individual play tactics, and overall game strategies are evaluated.

For our data let us consider results of the water polo tournament held at the city of Florence, Italy during September 1999 for the "European world Swimming Championships." A total of 12 national water polo teams competed with each other. Initially two round-robin groups with 6 teams in each group were formed. The best four teams from each group qualified for a paired round robin, and the winner teams were qualified for the semifinals, and so on.

i. The aim of the present study

1. to identify and interpret parameters as deviations of the attacking and respectfully defensive abilities from the average level in the tournament.

2. to predict and calculate the probabilities(win/draw/loss) of various potential future matches

3. generate and simulate a full virtual league based on our real tournament results and calculate witch position and with what probability each team will take

4. compare the team performance of elite European water polo teams according to their match and goals outcome during the tournament and understand what is the best team according to the data in attacking and defensive parameters

- *Our data are comprised of 44 tournament games from water euro 1999 along with goals of each time for each game.*
- *Poisson seems to be a realistic assumption for such data*

ii. Poisson model formulation (appendix model#1)

Here we consider the following formulation:

$$\begin{aligned} Y_{ij} &\sim \text{Poisson}(\lambda_{ik}) && \text{for } j = 1,2 \\ \log(\lambda_{i1}) &= \mu + a_{\text{HT}_i} + d_{\text{AT}_i} \\ \log(\lambda_{i2}) &= \mu + a_{\text{AT}_i} + d_{\text{HT}_i} \text{ for } i = 1,2,\dots,n, \end{aligned}$$

iii. Variable explanation

- The goals1 or goals2 of each team refers to the number of goals scored by each team within a game. Poisson seems to be a realistic assumption for such data. Usual scores are around 7.2 goals for each team, with a strong correlation between the scores of the competing teams.
- Here ht(i) and at(i) indicate the two opposing teams competing each other,
- N=Number of matches
- K=total teams
- a=attack parameter
- d=defense parameter
- the usual sum-to-zero constraints were imposed on both attacking and defensive parameters ak and dk

$$\sum_{k=1}^{K} a_k = 0 \text{ and } \sum_{k=1}^{K} d_k = 0$$

iv. Clarification

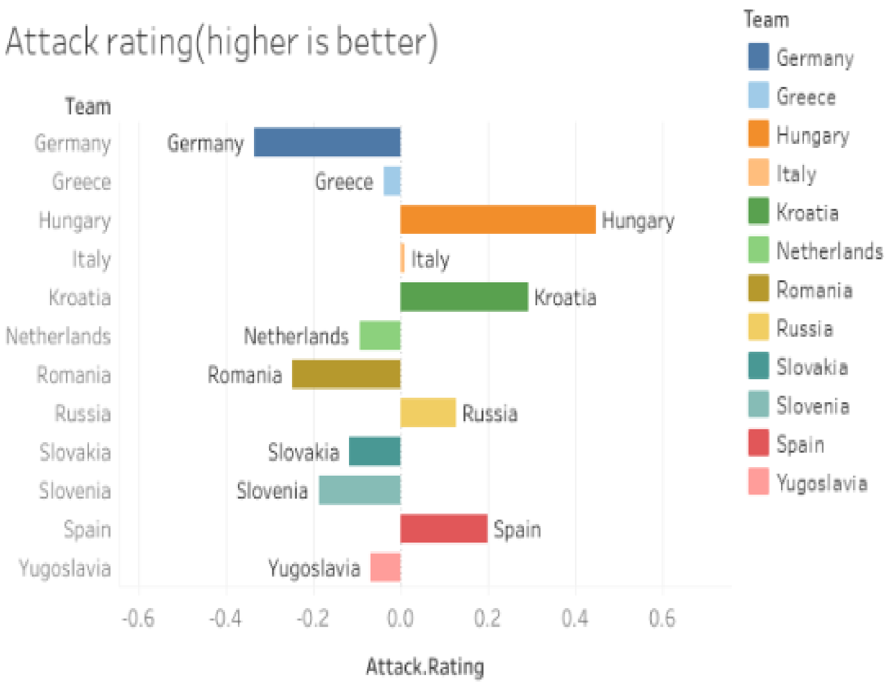
A positive attacking parameter in our results indicates that the team under consideration has an offensive performance that is better than the average level of the teams competing in the tournament. Likewise, a negative defensive parameter indicates that the team under consideration has a defensive performance that is better than the average level of the teams competing in the tournament in more simple words negative defense rating means the difficulty for the opponent to score a goal is higher than the average difficulty of the tournament(negative is better)

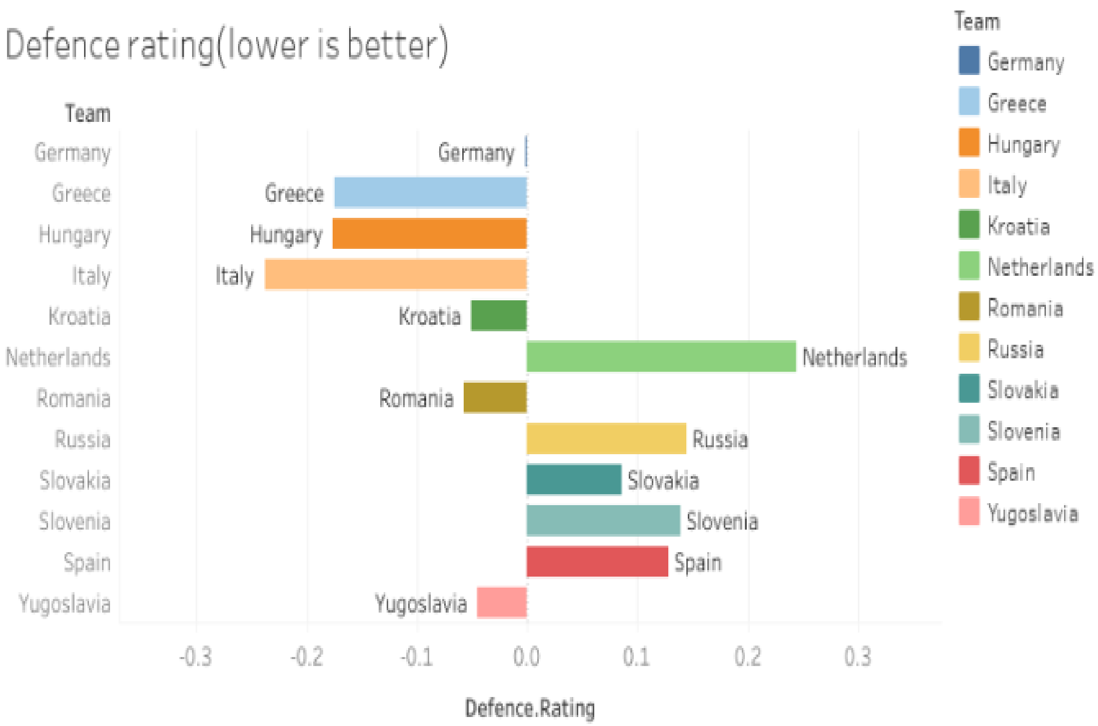
the posterior summaries results of our model for attacking and defensive parameters are the following:

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a[1]	-0.3533	0.1362	0.00663	-0.6001	-0.357	-0.07959	201	200
a[2]	-0.01617	0.1318	0.01188	-0.3038	-0.01103	0.2276	201	200
a[3]	0.4853	0.121	0.009252	0.2579	0.4877	0.7379	201	200
a[4]	0.04554	0.1383	0.01455	-0.2247	0.05867	0.2898	201	200
a[5]	0.3335	0.1379	0.01148	0.01476	0.3463	0.5561	201	200
a[6]	-0.1149	0.1637	0.01566	-0.4411	-0.1222	0.2154	201	200
a[7]	-0.3071	0.1639	0.01739	-0.6787	-0.2795	-4.88E-4	201	200
a[8]	0.1012	0.1404	0.01703	-0.1815	0.1071	0.3579	201	200
a[9]	-0.1014	0.1856	0.01643	-0.4381	-0.1071	0.2405	201	200
a[10]	-0.1579	0.1728	0.01608	-0.5201	-0.1692	0.2096	201	200
a[11]	0.172	0.1106	0.0147	-0.01994	0.1786	0.3875	201	200
a[12]	-0.08663	0.128	0.01257	-0.3475	-0.07106	0.1241	201	200
d[1]	0.02069	0.1283	0.008824	-0.2335	0.01234	0.2631	201	200
d[2]	-0.1742	0.1436	0.01342	-0.4734	-0.1631	0.07576	201	200
d[3]	-0.2161	0.1463	0.01631	-0.4906	-0.2203	0.08058	201	200
d[4]	-0.2579	0.1488	0.01771	-0.5806	-0.2398	0.01363	201	200
d[5]	-0.05909	0.1364	0.0168	-0.366	-0.05684	0.1717	201	200
d[6]	0.2723	0.1302	0.01436	-0.009089	0.2848	0.5063	201	200
d[7]	-0.03317	0.1653	0.01746	-0.3304	-0.02827	0.2956	201	200
d[8]	0.1914	0.137	0.01712	-0.09547	0.1881	0.4678	201	200
d[9]	0.03899	0.1468	0.01485	-0.2556	0.03092	0.3172	201	200
d[10]	0.1118	0.1298	0.01348	-0.1756	0.1304	0.3495	201	200
d[11]	0.1484	0.1374	0.01581	-0.1328	0.1623	0.4243	201	200
d[12]	-0.04313	0.1272	0.01406	-0.2926	-0.03308	0.2078	201	200

1.Germany//2.Greece//3.Hungary//4.Italy//5Kroatia//6.Netherlands.//7.Romania//8.Russia//9.Slovakia//10.Slovenia//11.Spain//12.Yugoslavia

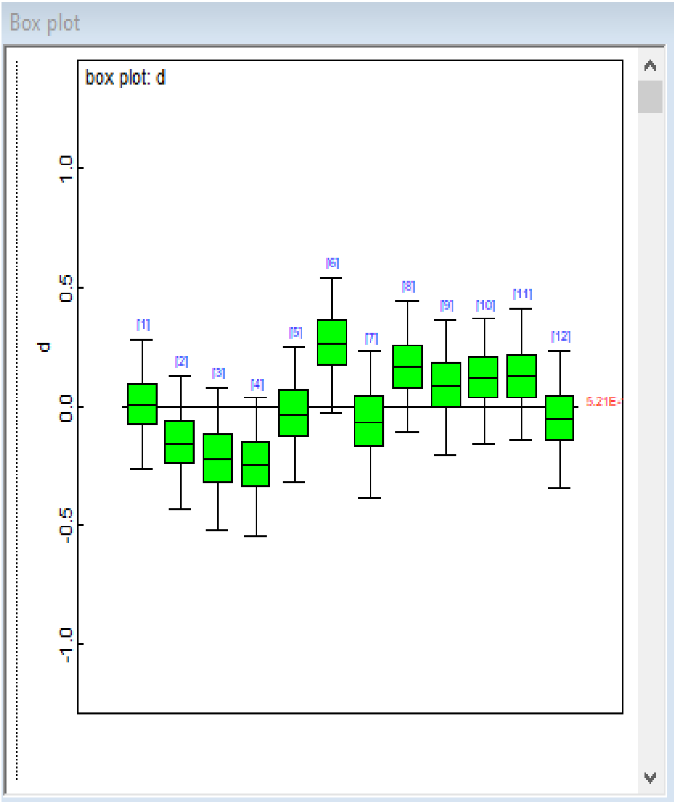


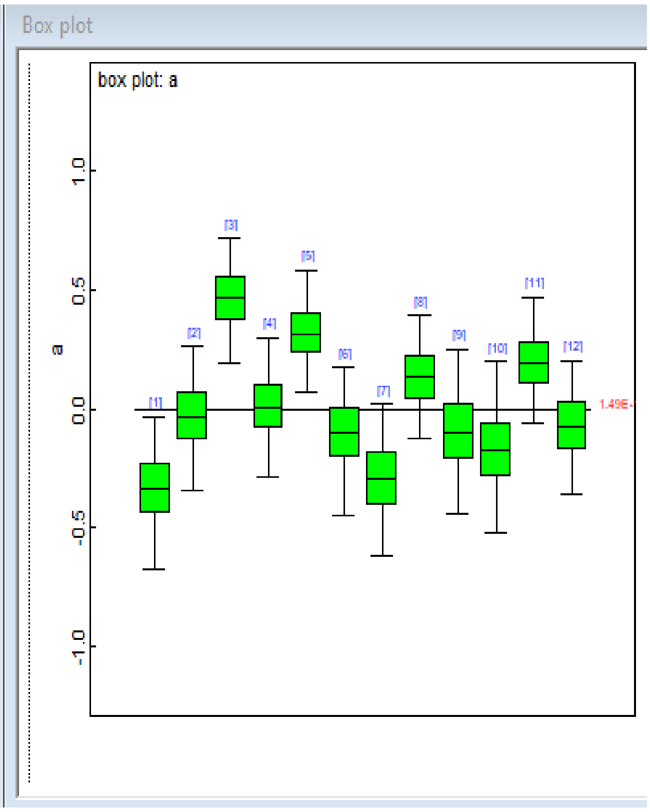


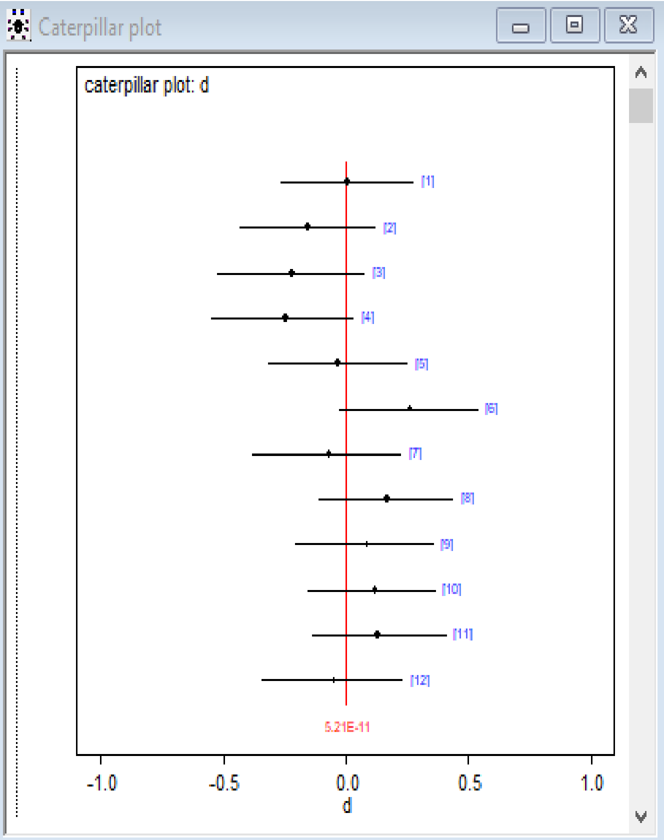
- As we can see the teams with the best attacking rating according to our model are: **Croatia, Russia, Spain, Hungary.**
- Teams with best defensive rating are: **Greece, Italy, Hungary, Croatia, Yugoslavia.**

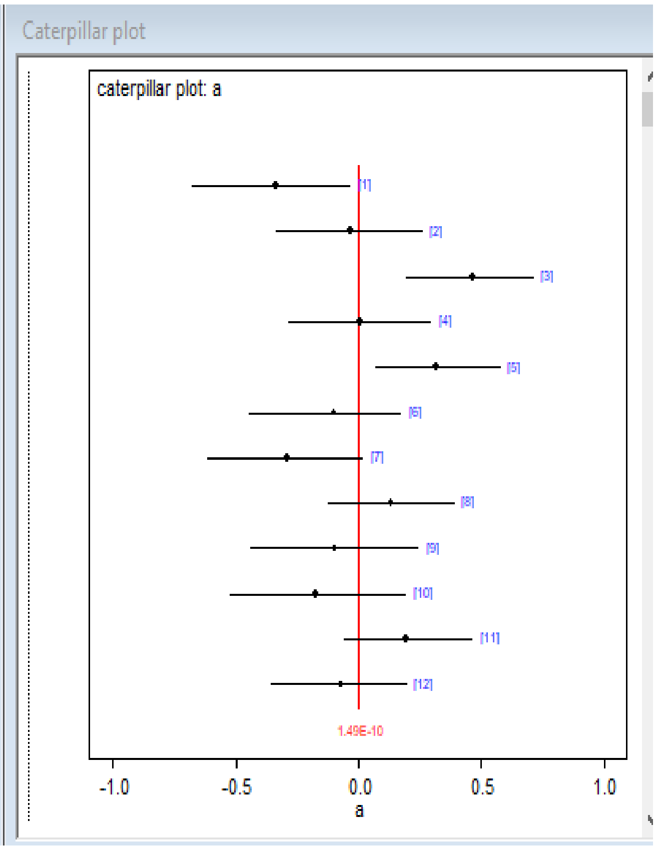
As we can see both our finalists Hungary and Kroatia have both excellent attacking and defensive abilities compare to the other teams

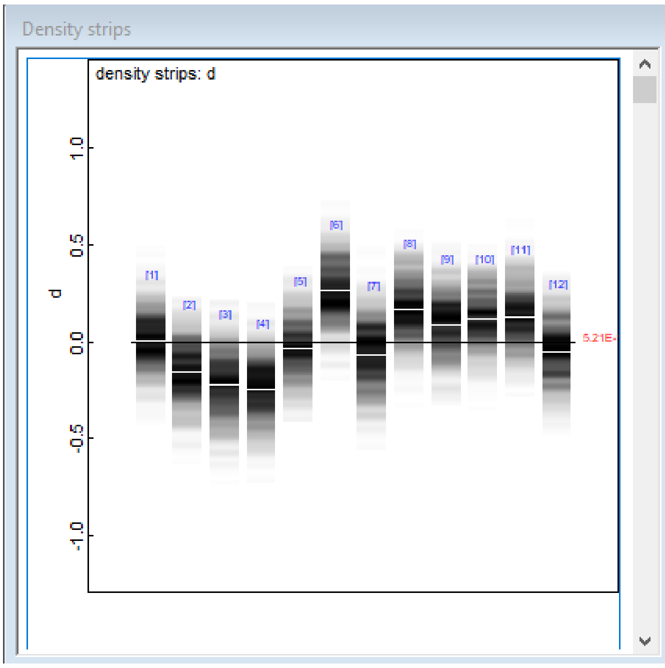
<u>Here are also WinBugs compare tools to check our parameters</u>

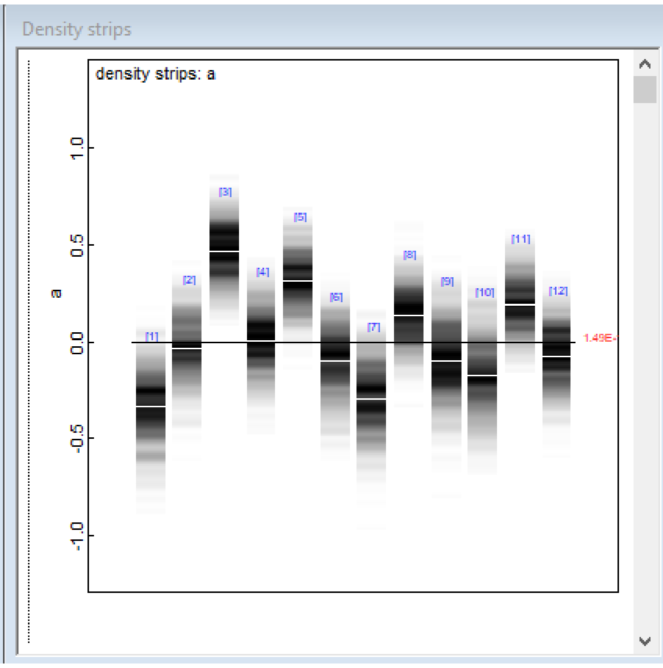












Is the poisson model thought the best distribution to model our data?

Although the Binomial or Negative Binomial have been proposed in the late 1970s (Pollard et al. 1977), the Poisson distribution has been widely accepted as a suitable model for these quantities; in particular, a simplifying assumption often used is that of independence between the goals scored by the home and the away team. For instance, Maher (1982) used a model with two independent Poisson variables where the relevant parameters are constructed as the product of the strength in the attack for one team and the weakness in defense for the other. Despite that, some authors have shown empirical, although relatively low, levels of correlation between the two quantities (**Lee 1997, Karlis & Ntzoufras 2000**). Consequently, the use of more sophisticated models have been proposed, for instance by Dixon & Coles (1997), who applied a correction factor to the independent Poisson model to improve the performance in terms of prediction. More recently, **Karlis & Ntzoufras** (2000, 2003) advocated the use of a bivariate Poisson distribution that has a more complicated formulation for the likelihood function and includes an additional parameter explicitly accounting for the covariance between the goals scored by the two competing teams**. We should check thought DIC criteria to confirm ourselves which distribution model our data better**

v. Negative binomial model

Here we consider the following formulation:

$Y_i \sim NB(\pi_{G_i+1}, r_{G_i+1})$ for $i = 1,2,\ldots,44$

With prior distributions
$\pi_j \sim U(0,1)$ and $r_j \sim \text{gamma}(0.001,0.001)$

Attacking and Defensive parameters for NB distribution

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a[1]	-0.3414	0.1618	0.00753	-0.6729	-0.3361	-0.03598	1001	1000
a[2]	-0.009488	0.1388	0.01218	-0.2958	0.006145	0.2268	1001	1000
a[3]	0.4674	0.1229	0.01073	0.2073	0.4923	0.6886	1001	1000
a[4]	-0.002057	0.14	0.01384	-0.2523	-0.01307	0.2772	1001	1000
a[5]	0.3219	0.1267	0.01286	0.0574	0.335	0.5492	1001	1000
a[6]	-0.08918	0.1674	0.01583	-0.3938	-0.1036	0.2318	1001	1000
a[7]	-0.2777	0.1807	0.01833	-0.6284	-0.2674	0.08979	1001	1000
a[8]	0.1301	0.1459	0.01736	-0.1409	0.1129	0.4178	1001	1000
a[9]	-0.1155	0.1847	0.01957	-0.4806	-0.1072	0.2208	1001	1000
a[10]	-0.1878	0.1761	0.01723	-0.5135	-0.197	0.1529	1001	1000
a[11]	0.1593	0.1256	0.01058	-0.09336	0.165	0.4188	1001	1000
a[12]	-0.05556	0.1349	0.01214	-0.3422	-0.05358	0.2246	1001	1000
d[1]	0.002977	0.1408	0.009772	-0.2854	0.004258	0.2723	1001	1000
d[2]	-0.137	0.1436	0.01549	-0.4396	-0.1241	0.136	1001	1000
d[3]	-0.1924	0.1507	0.01795	-0.5084	-0.1813	0.07121	1001	1000
d[4]	-0.2501	0.1573	0.0162	-0.5282	-0.256	0.1181	1001	1000
d[5]	-0.06213	0.125	0.01311	-0.3406	-0.05443	0.1478	1001	1000
d[6]	0.2905	0.1449	0.01645	-0.003153	0.2841	0.5612	1001	1000
d[7]	-0.06975	0.1701	0.01781	-0.384	-0.06004	0.2317	1001	1000
d[8]	0.1764	0.1432	0.0152	-0.118	0.1964	0.4524	1001	1000
d[9]	0.07639	0.1508	0.01598	-0.2888	0.07483	0.3454	1001	1000
d[10]	0.1059	0.1371	0.01484	-0.144	0.1009	0.356	1001	1000
d[11]	0.1077	0.1407	0.01652	-0.1414	0.1012	0.3889	1001	1000
d[12]	-0.04844	0.1446	0.01431	-0.3545	-0.03156	0.2305	1001	1000

As we can see the values of the parameters are pretty similar with our poisson model with small differences

Our DIC value though is the following

For the **Negative binomial** Model:

Deviance information

	Dbar	Dhat	DIC	pD
goals1	197.1	185.3	209.0	11.81
goals2	198.9	187.1	210.8	11.86
total	396.1	372.4	419.8	23.67

For the **poisson** Model:

Deviance information

	Dbar	Dhat	DIC	pD
goals1	196.3	184.7	207.9	11.6
goals2	198.4	186.9	209.9	11.5
total	394.7	371.6	417.8	23.11

As we could suspect the DIC value(smaller is better) of Poisson model is better than the negative binomial model so we will continue our analysis with our Poisson distribution

2. PREDICT MATCH RESULTS WITH PROPABILITIES LIKE IN ONLINE BETTING

As we can see below these are the actual ranking results in our tournament

RANK	TEAM
	Hungary
	Croatia
	Italy
4.	Greece
5.	Russia
6.	Spain
7.	FR Yugoslavia
8.	Germany
9.	Romania
10.	Slovakia
11.	Slovenia
12.	Netherlands

But what were the actual probabilities of each match before it was played according to our model. Our purpose here is to replay the semifinals and finals to see if the results will be different.

Our model(appendix model#2+3)

Lets play again the semifinals and our finals

Results semifinals

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
goals1[39]	5.638	2.685	0.1671	1.0	5.0	11.0	501	500
goals1[40]	9.016	3.545	0.1359	3.0	9.0	17.0	501	500
goals2[39]	6.63	2.734	0.1305	2.0	7.0	13.0	501	500
goals2[40]	5.282	2.464	0.175	1.0	5.0	11.0	501	500
outcome[1,1]	0.342	0.4744	0.0268	0.0	0.0	1.0	501	500
outcome[1,2]	0.094	0.2918	0.01161	0.0	0.0	1.0	501	500
outcome[1,3]	0.564	0.4959	0.02793	0.0	1.0	1.0	501	500
outcome[2,1]	0.778	0.4156	0.0199	0.0	1.0	1.0	501	500
outcome[2,2]	0.068	0.2517	0.01413	0.0	0.0	1.0	501	500
outcome[2,3]	0.154	0.3609	0.01537	0.0	0.0	1.0	501	500
pred.diff[1]	-0.992	3.672	0.2065	-8.0	-1.0	6.0	501	500
pred.diff[2]	3.734	4.323	0.2395	-4.0	4.0	13.0	501	500

Before we start analyzing we should know that average expected goals per team in each match was **7.2 goals**

Semifinal 1: Greece-Croatia

Win Greece **0.342**

draw **0.094**

Win Croatia **0.564**

before the semis it was the most favorable

Expected goals for Greece were **5.638** and for Croatia **6.63**. Both are below average because of the very good defensive capabilities of the both teams despite that the the actual score was 7-10 with **Croatia winner**

Semifinal 2: Hungary-Italy

Win Hungary 0.778

draw 0.068

Win Italy 0.154.

Hungary is the absolute favorable with 9 expected scoring goals against

Italy was the outsider with only 5.2 expected goals.

Hungary was the winner with the score of 7-5 only 2 goals diff vs the 3.8 expected. This means Italy did a good job in this match despite the lose.

Results finals

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
goals1[41]	8.374	3.18	0.1804	3.0	8.0	15.0	501	500
goals2[41]	6.376	2.949	0.1339	1.0	6.0	14.0	501	500
outcome[1,1]	0.648	0.4776	0.02137	0.0	1.0	1.0	501	500
outcome[1,2]	0.08	0.2713	0.009883	0.0	0.0	1.0	501	500
outcome[1,3]	0.272	0.445	0.01976	0.0	0.0	1.0	501	500

Final: Hungary-Croatia

Win Hungary **0.648**

draw **0.08**

Win Croatia **0.272**

As we can see the big favorite is Hungary again.Expected goals for Hungary were **8.374** and for Croatia **6.37**. As we can see despite the great attacking ability of Kroatia the expected goals for them were below the 7.2 average mark pointing the excellent defensive and attacking capability of Hungary again!

Actual final score was 15-12 with **Hungary winner.**

Results: we predicted correct in all 3 matches without counting at our model the semifinal and final games so they cant effect our predicted results

***Conclusion**: maybe I should start betting more money in sports matches*

3. WATER POLO LEAGUE SIMULATION

We can generate a new simulated tournament with these teams.

What will be the expected results if all teams played against eachother two times?

will the results change between our simulated tournament and our real tournament results?

what is the actual probabilities for each team to concuer each position?

Win is 3 points

Draw is 1 point

Lose is 0 points

Team with most points are the champion.

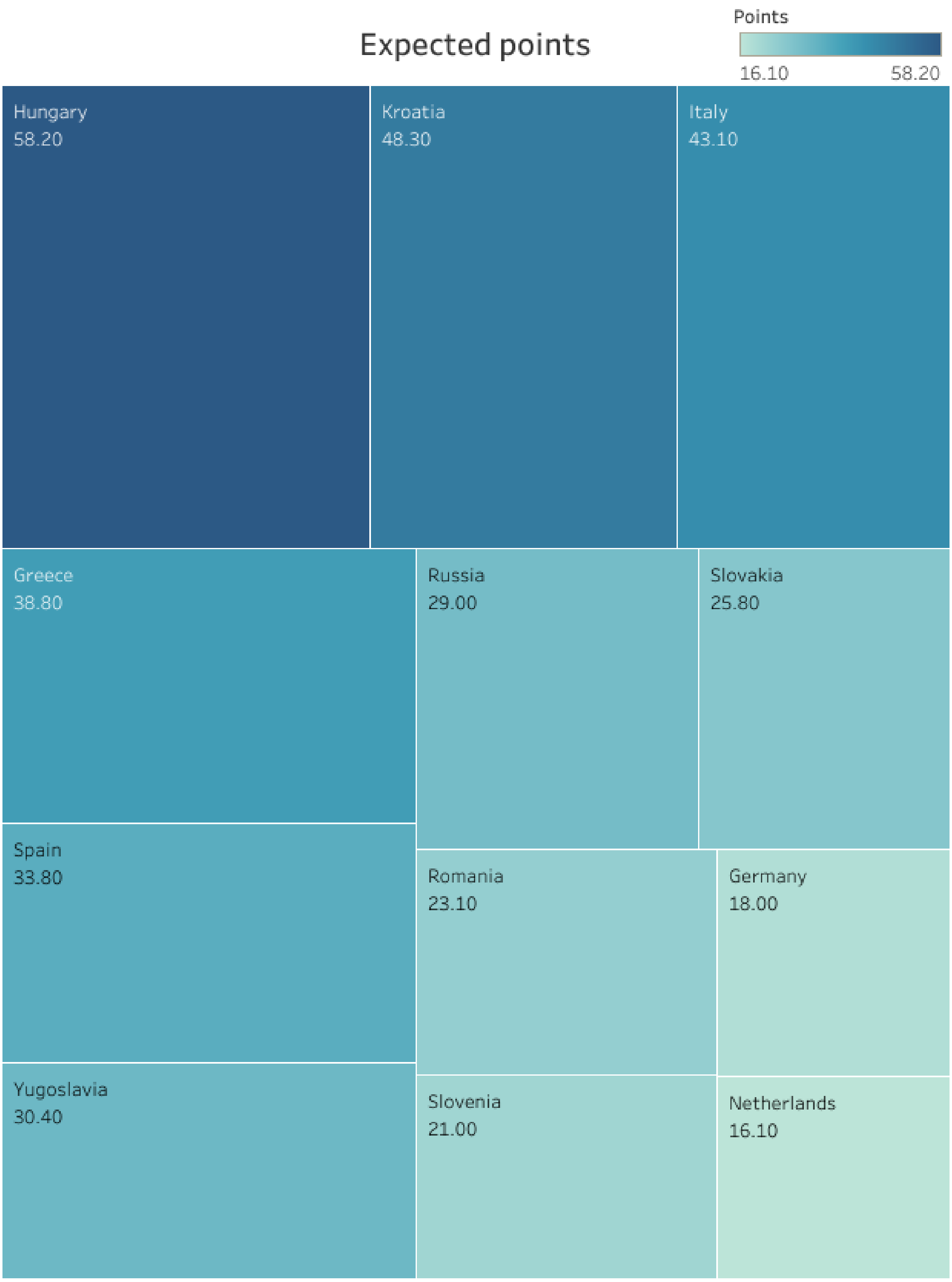
League simulation modelling (appendix model#4)

Results

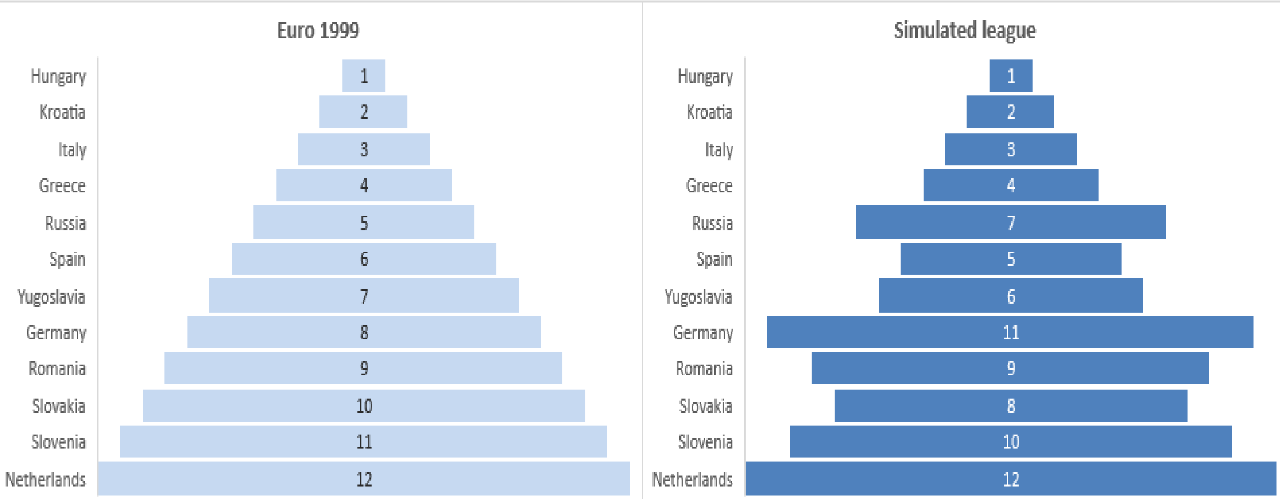
Node statistics

ranks[1]	9.714	2.14	0.09501	5.0	10.0	12.0	101	500
ranks[2]	4.732	2.232	0.1723	1.0	4.0	10.0	101	500
ranks[3]	1.346	0.8163	0.04234	1.0	1.0	3.0	101	500
ranks[4]	3.804	2.007	0.1304	1.0	3.0	9.0	101	500
ranks[5]	2.752	1.516	0.08011	1.0	2.0	7.0	101	500
ranks[6]	10.06	2.015	0.1065	5.0	11.0	12.0	101	500
ranks[7]	8.378	2.413	0.1585	4.0	9.0	12.0	101	500
ranks[8]	7.018	2.465	0.1443	2.0	7.0	12.0	101	500
ranks[9]	7.77	2.489	0.155	3.0	8.0	12.0	101	500
ranks[10]	8.978	2.472	0.1489	4.0	9.0	12.0	101	500
ranks[11]	5.782	2.461	0.1618	1.0	6.0	11.0	101	500
ranks[12]	6.634	2.398	0.1509	2.0	7.0	11.0	101	500
total.points[1]	18.09	8.415	0.3978	4.0	17.0	37.0	101	500
total.points[2]	38.81	9.426	0.7141	20.0	40.0	55.0	101	500
total.points[3]	58.2	5.807	0.3476	44.0	60.0	66.0	101	500
total.points[4]	43.16	9.174	0.6195	23.0	44.0	59.0	101	500
total.points[5]	48.34	8.03	0.4638	31.0	49.0	61.0	101	500
total.points[6]	16.09	8.238	0.5445	3.0	15.0	34.0	101	500
total.points[7]	23.1	9.383	0.6279	6.0	23.0	41.0	101	500
total.points[8]	28.99	9.311	0.579	12.0	28.0	48.0	101	500
total.points[9]	25.81	9.774	0.5953	8.0	26.0	45.0	101	500
total.points[10]	21.05	9.917	0.5622	5.0	20.0	42.0	101	500
total.points[11]	33.82	9.627	0.6226	16.0	34.0	52.0	101	500
total.points[12]	30.4	9.065	0.5605	13.0	30.0	47.0	101	500

Total points expected average points



As we can from before our actual ranking from the tournament vs our simulated league is the following

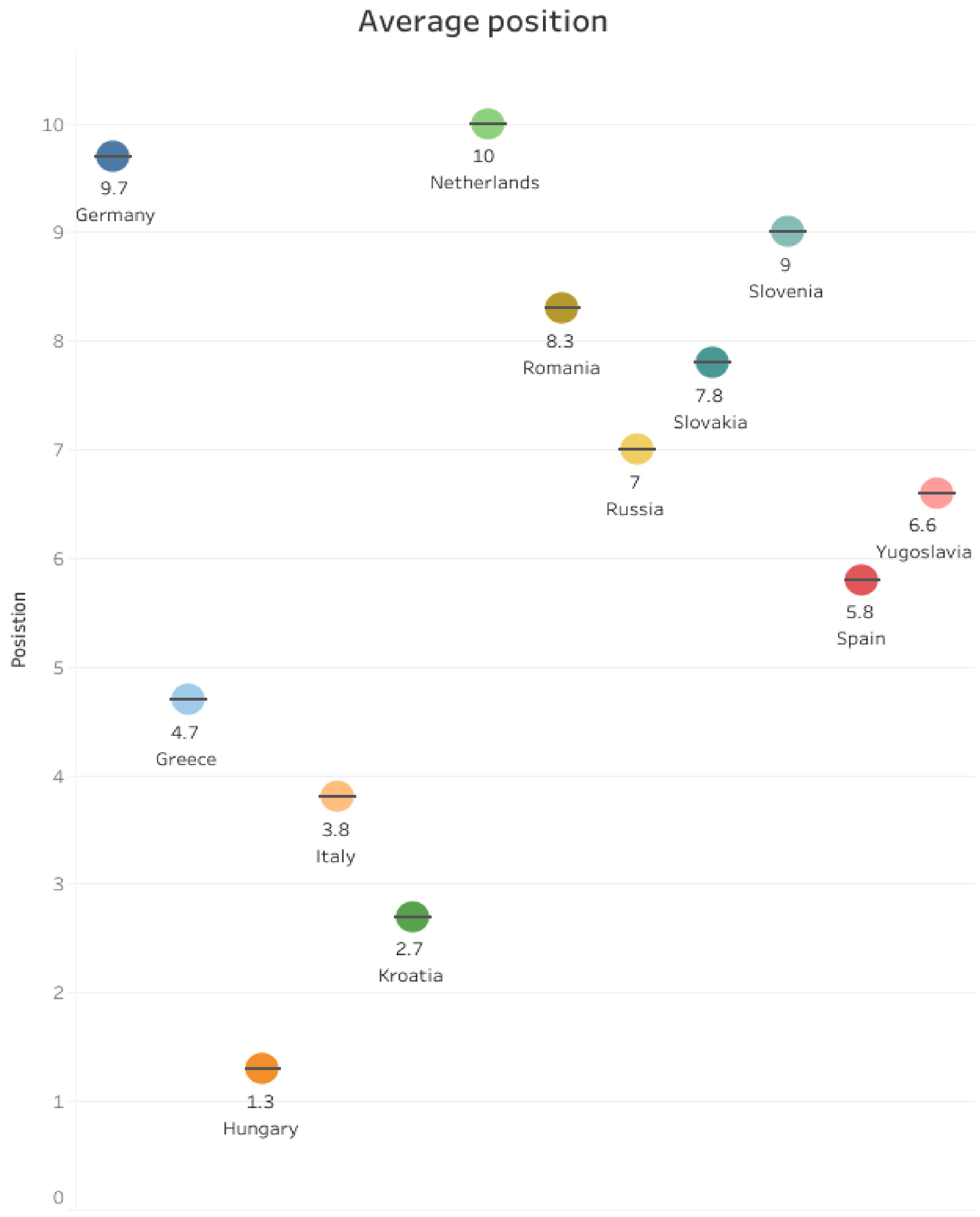


The overall differences are similar which means the actual tournament position were almost identical to our generated league. This translates that our total points, attacking and defensive parameters were accurate and translated into actual accurate results

Biggest differences

- Top 4 teams are exactly the same
- Russia went from 5th position to 7th
- Spain and Yugoslavia were predicted 1 place higher
- Biggest changes are Germany from 8th position went to 11th and Slovakia from 10th went up to 8th place

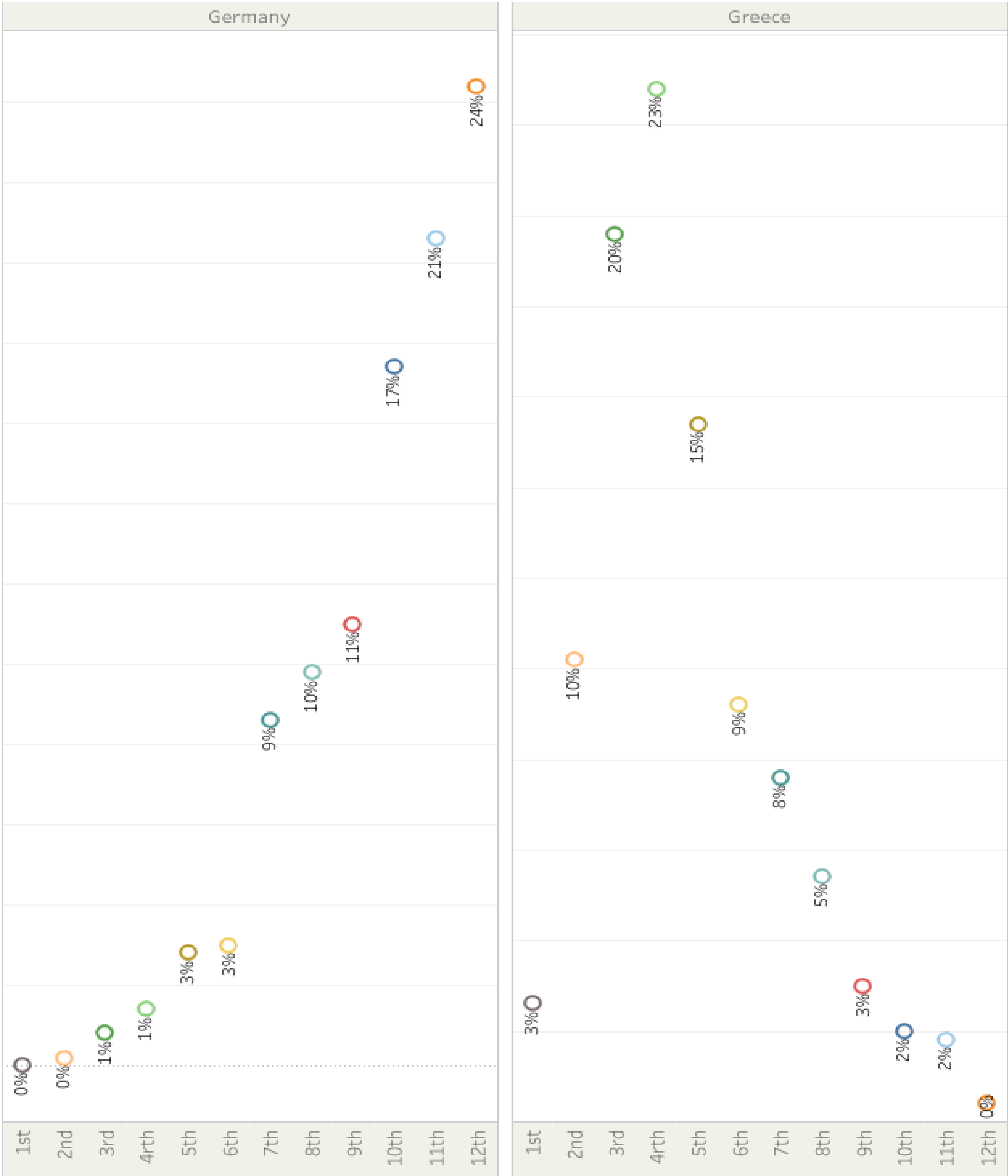
vi. Average predictive position for each team



Here we can see the average position of each team for our generated water polo league.

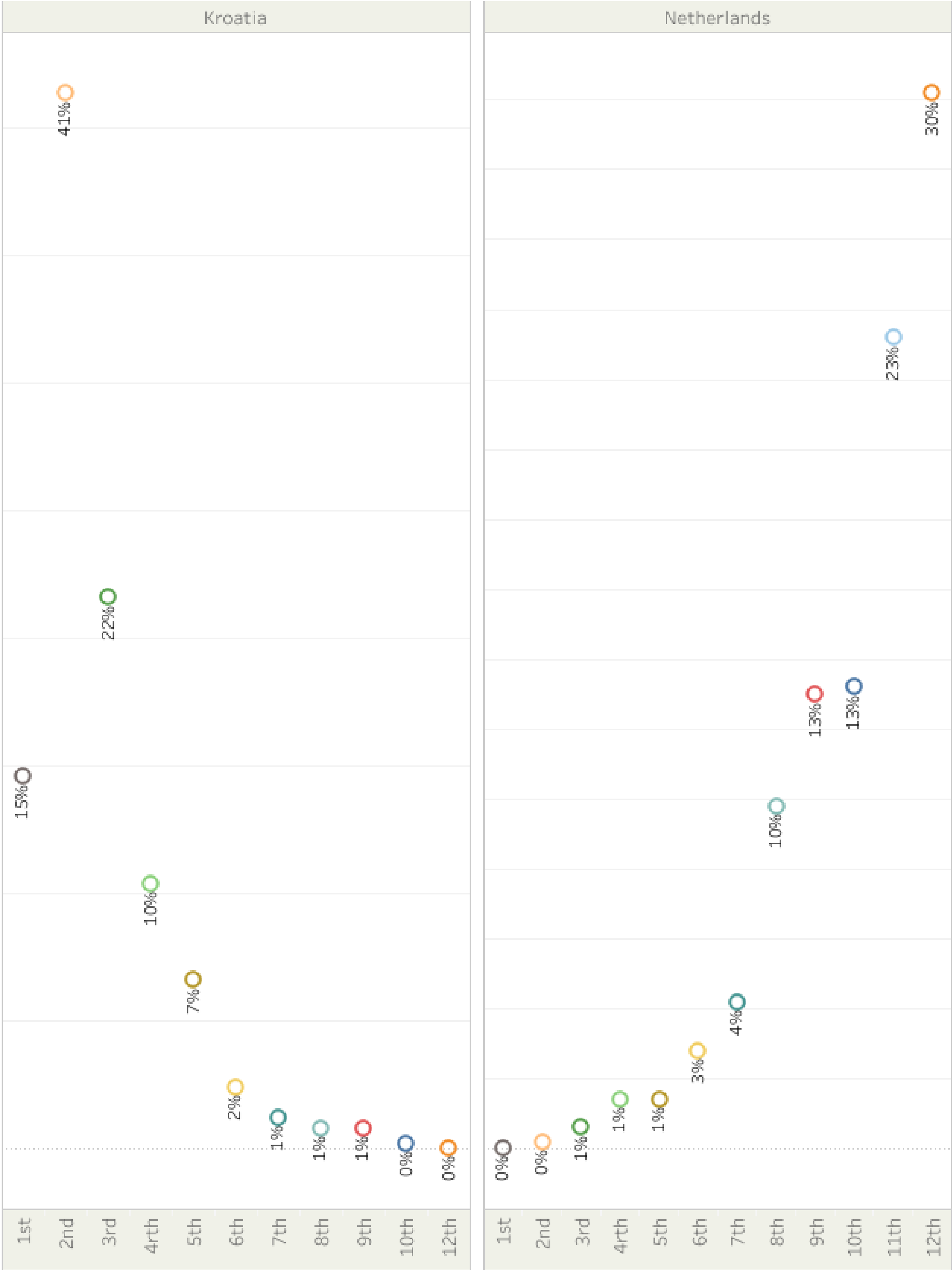
vii. Team ranking probabilities for each position

Here we can see each probability for each team for every position in our virtual league

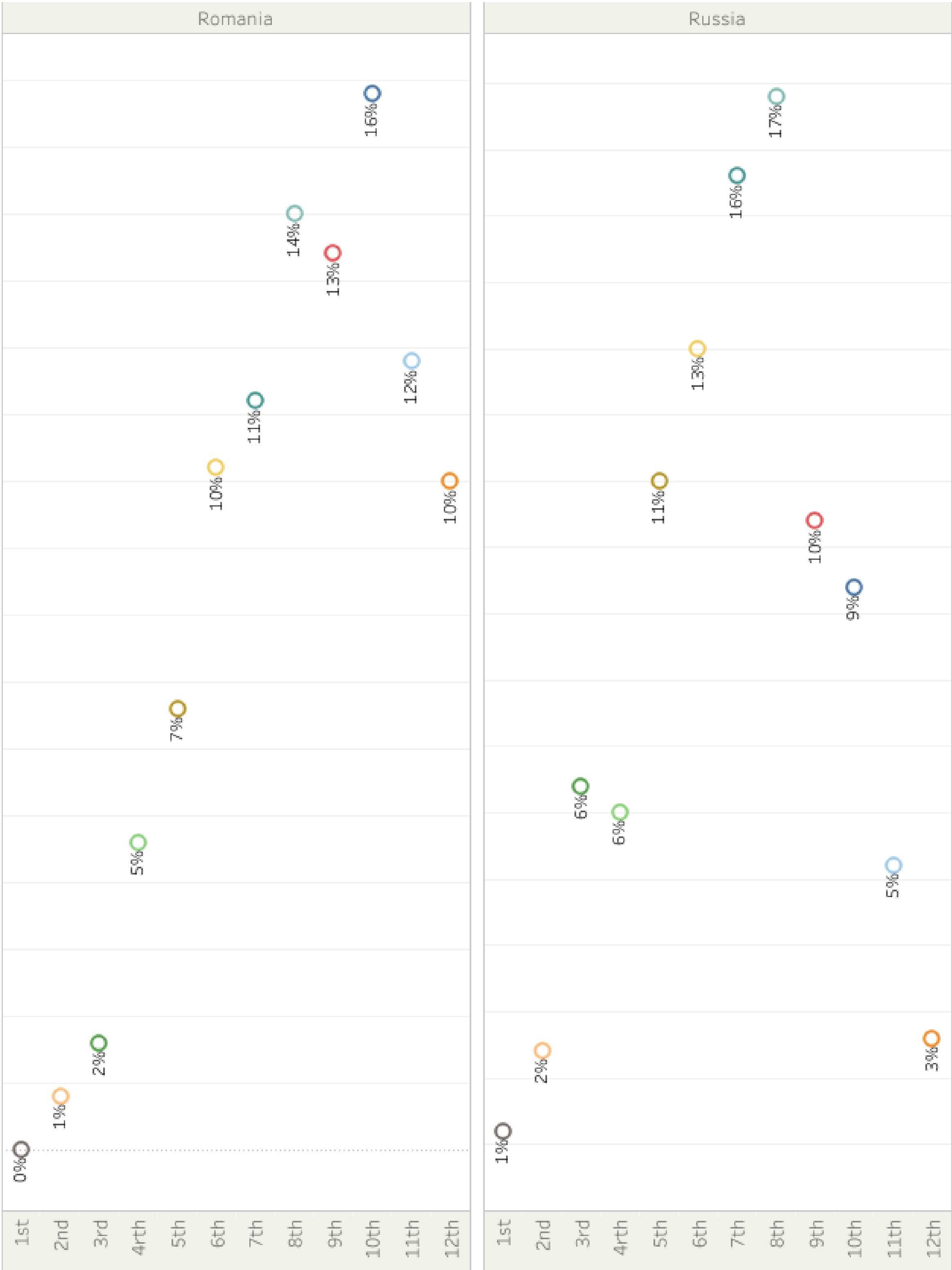


- **Germany**: One of our lowest teams with 21% for 11th place and 24% for last place
- **Greece**: One of the best teams with 20% for 3rd and 23% for 4th place

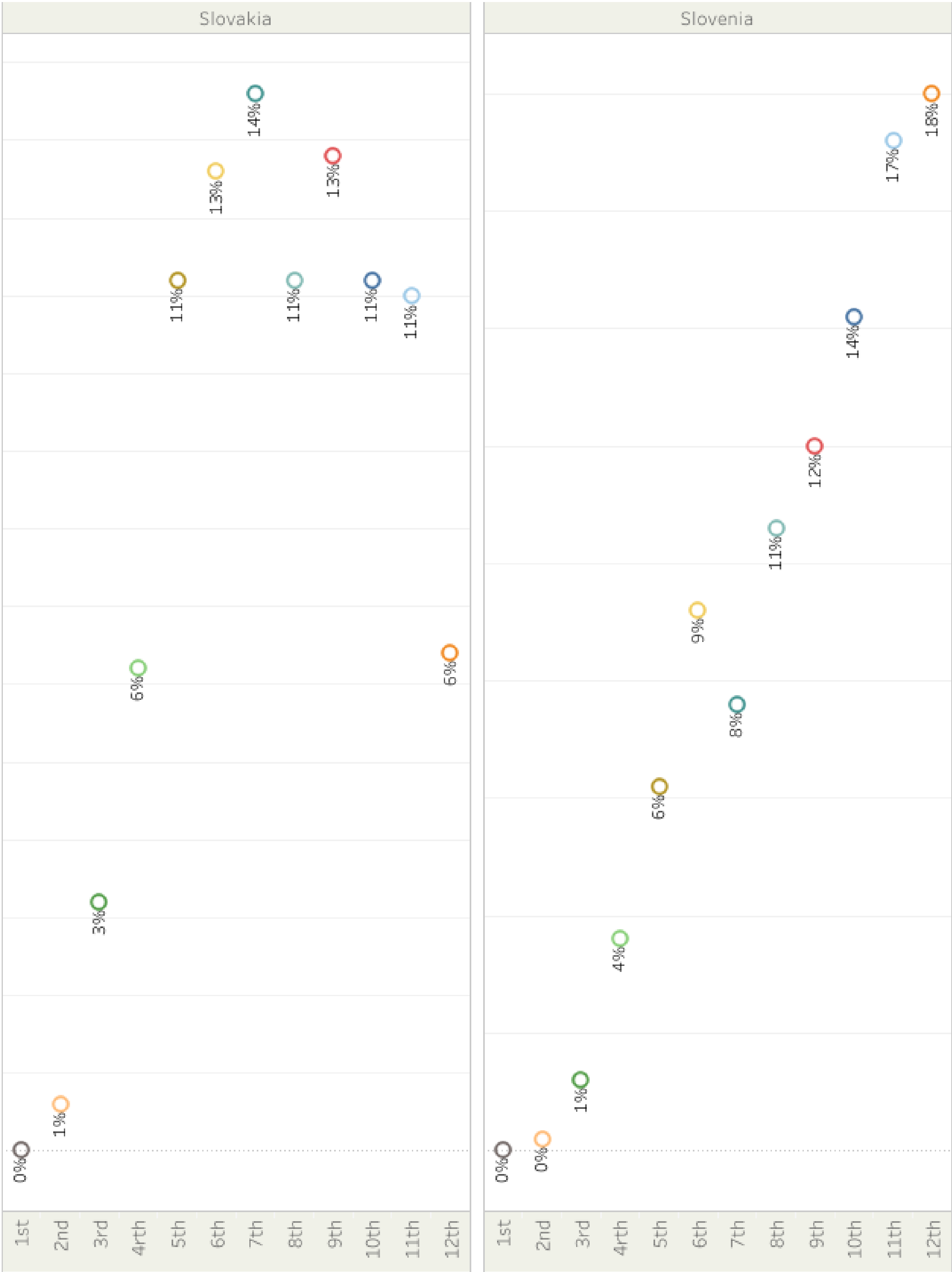
- **Hungary**: our best team by far with 77% for the first place and 16% for 2^{nd}
- **Italy**: one of our best team with 23% and 22% for 2^{nd} and 3^{rd} place accordingly.



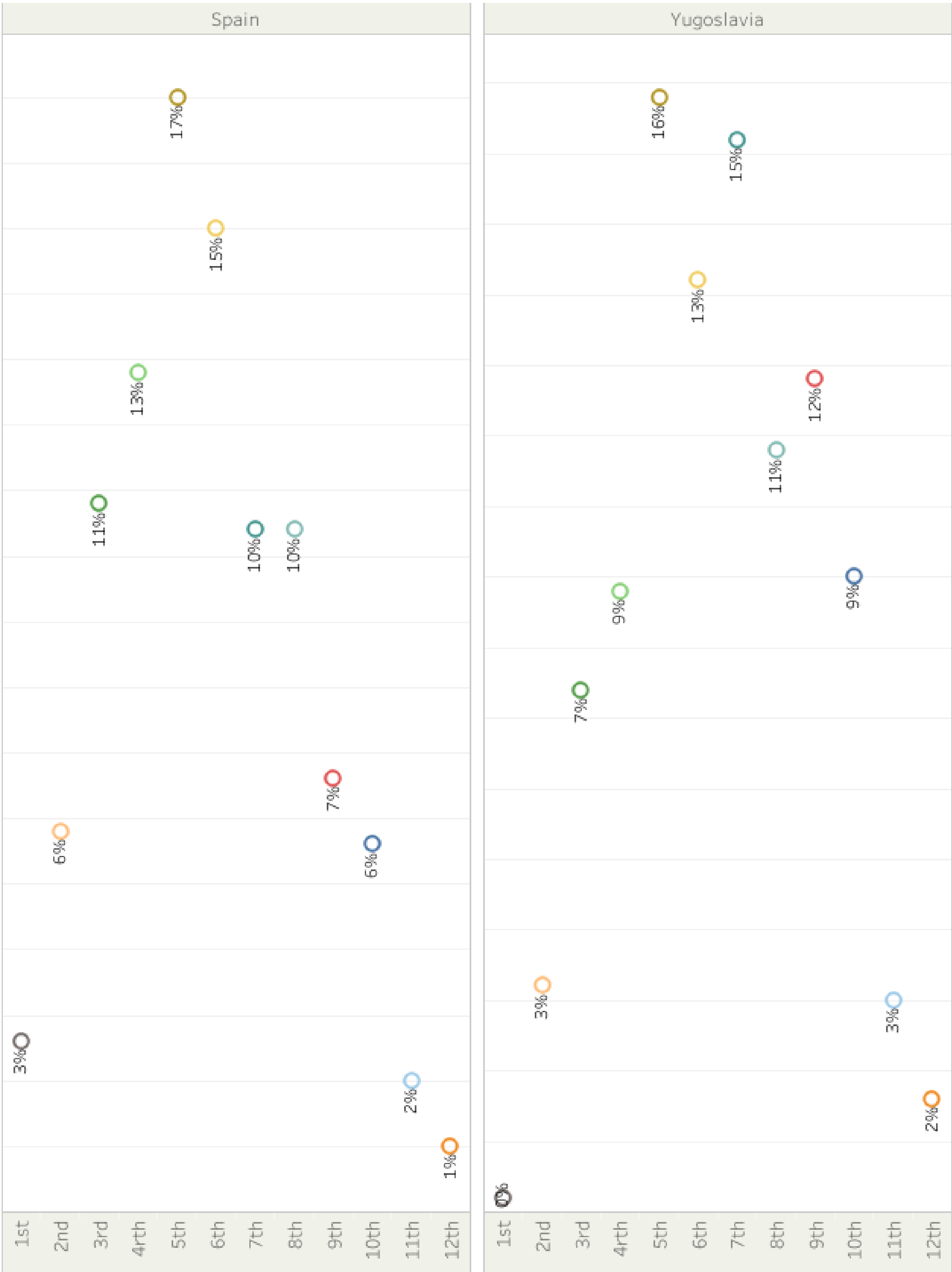
- **Croatia**: one of the best team with 15% for 1st and 22% for 2nd place
- **Netherlands**: one of our weakest team with 23% for last place



- **Romania**: one of our weakest team with 14% for 8th and 16% for 10th place
- **Russia**: one of our average team with 16% and 17% for 7th and 8th place accordingly.



- **Slovakia**: one of our average team despite the real low tournament ranking with 14% and 13% for 6th and 5th place
- **Slovenia**: one of our weakest team with 18% for last place and 17% for 11th



- **Spain**: one of our strongest team despite the 6th real tournament final ranking with 11% for 4rd and 17% for 5th
- **Yugoslavia**: one of our better average teams with 16% for 5th and 15% for 7th

4. ADD EFFECTS TO OUR MODEL

In our previous models we conspired our models with no additional effects

What will happen if we add random effect elements in our games?

What will happen if we use fixed effects model?

What is the difference between random and fixed effects?

Fixed effects

Fixed effect models assume that the explanatory variable has a fixed or constant relationship with the response variable across all observations

Random effects

A random-effects model assumes that explanatory variables have fixed relationships with the response variable across all observations, but that these fixed effects may vary from one observation to another.

Differences

In the linear model, each level of a fixed effect contributes a fixed amount to the expected value of the dependent variable. What makes a random effect different is that each level of a random effect contributes an amount that is viewed as a sample from a population of normally distributed variables, each with mean 0, and an unknown variance

A fixed-effects model supports prediction about only the levels/categories of features used for training. A random-effects model, by contrast, allows predicting something about the population from which the sample is drawn.

viii. Fixed vs random effect for our data

Game fixed effect model

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a[1]	-0.338	0.1618	0.003973	-0.6778	-0.3348	-0.0322	1001	1000
a[2]	-0.03479	0.1495	0.00831	-0.3392	-0.03531	0.2621	1001	1000
a[3]	0.4638	0.1315	0.007771	0.1956	0.4692	0.7154	1001	1000
a[4]	0.007653	0.1474	0.005711	-0.2849	0.01088	0.2971	1001	1000
a[5]	0.3149	0.13	0.005603	0.0726	0.3118	0.5799	1001	1000
a[6]	-0.1011	0.1573	0.006607	-0.4447	-0.0873	0.1728	1001	1000
a[7]	-0.2921	0.1632	0.007926	-0.6141	-0.2887	0.01902	1001	1000
a[8]	0.1324	0.1329	0.005511	-0.1236	0.1382	0.3918	1001	1000
a[9]	-0.09882	0.1707	0.008736	-0.4414	-0.09999	0.2481	1001	1000
a[10]	-0.1752	0.1787	0.0106	-0.5234	-0.1827	0.1961	1001	1000
a[11]	0.1947	0.1312	0.007064	-0.0596	0.1931	0.4662	1001	1000
a[12]	-0.07341	0.1436	0.007749	-0.3553	-0.07138	0.1983	1001	1000

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
d[1]	0.006048	0.1336	0.00329	-0.2624	0.005408	0.2787	1001	1000
d[2]	-0.1538	0.1414	0.006997	-0.4338	-0.1551	0.1258	1001	1000
d[3]	-0.2219	0.1551	0.007933	-0.5219	-0.2189	0.07874	1001	1000
d[4]	-0.2465	0.1481	0.009161	-0.5481	-0.2407	0.03418	1001	1000
d[5]	-0.03111	0.1443	0.007043	-0.3149	-0.03305	0.2517	1001	1000
d[6]	0.2613	0.1448	0.007402	-0.02893	0.2542	0.5411	1001	1000
d[7]	-0.06797	0.1579	0.009415	-0.3842	-0.06313	0.2288	1001	1000
d[8]	0.1666	0.1414	0.00646	-0.1088	0.1621	0.4426	1001	1000
d[9]	0.08769	0.1426	0.006295	-0.2061	0.09408	0.3599	1001	1000
d[10]	0.12	0.1339	0.006386	-0.1536	0.1197	0.366	1001	1000
d[11]	0.1286	0.1405	0.005656	-0.1361	0.1249	0.4117	1001	1000
d[12]	-0.04894	0.1461	0.007044	-0.3414	-0.04529	0.2352	1001	1000

And our DIC

Deviance information

	Dbar	Dhat	DIC	pD
g	394.8	371.6	418.0	23.21
game	29450.0	29450.0	29450.0	-3.388E-11
phase	1281.0	1281.0	1281.0	3.757E-13
total	31130.0	31100.0	31150.0	23.21

Game Random effect model

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a[1]	-0.3404	0.1621	0.003384	-0.6621	-0.3376	-0.02796	1001	2000
a[2]	-0.02875	0.149	0.005998	-0.3356	-0.02852	0.2516	1001	2000
a[3]	0.4558	0.1277	0.004565	0.2043	0.4572	0.7073	1001	2000
a[4]	0.004502	0.1487	0.005065	-0.2908	0.006756	0.301	1001	2000
a[5]	0.3219	0.1404	0.005487	0.03711	0.321	0.5858	1001	2000
a[6]	-0.1109	0.1579	0.005299	-0.423	-0.1118	0.1911	1001	2000
a[7]	-0.2773	0.1676	0.006498	-0.6152	-0.274	0.03126	1001	2000
a[8]	0.1188	0.1408	0.005416	-0.1613	0.1204	0.3872	1001	2000
a[9]	-0.08972	0.1758	0.007478	-0.4444	-0.08693	0.252	1001	2000
a[10]	-0.1713	0.1704	0.007039	-0.5122	-0.1703	0.1592	1001	2000
a[11]	0.1883	0.1323	0.005575	-0.07616	0.1906	0.4383	1001	2000
a[12]	-0.07083	0.143	0.004951	-0.3494	-0.06764	0.2058	1001	2000

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
d[1]	0.005781	0.1377	0.002909	-0.2733	0.008618	0.2698	1001	2000
d[2]	-0.1654	0.142	0.005485	-0.4474	-0.1645	0.1117	1001	2000
d[3]	-0.2229	0.1614	0.006185	-0.5453	-0.2166	0.08379	1001	2000
d[4]	-0.2391	0.1454	0.005726	-0.5434	-0.2354	0.03715	1001	2000
d[5]	-0.02896	0.1383	0.005161	-0.3088	-0.02908	0.249	1001	2000
d[6]	0.2637	0.1526	0.00526	-0.03032	0.2656	0.5424	1001	2000
d[7]	-0.06642	0.1668	0.007342	-0.39	-0.0655	0.2453	1001	2000
d[8]	0.1653	0.1401	0.00628	-0.1208	0.1678	0.4326	1001	2000
d[9]	0.07635	0.158	0.005886	-0.2489	0.07955	0.3686	1001	2000
d[10]	0.1237	0.1504	0.006131	-0.1831	0.1282	0.403	1001	2000
d[11]	0.1318	0.1453	0.005079	-0.1468	0.1342	0.4203	1001	2000
d[12]	-0.04372	0.1419	0.004887	-0.3283	-0.04048	0.2215	1001	2000

Deviance information

	Dbar	Dhat	DIC	pD
g	392.4	366.3	418.5	26.09
game	29450.0	29450.0	29450.0	-3.388E-11
phase	1281.0	1281.0	1281.0	3.757E-13
total	31120.0	31100.0	31150.0	26.09

Number of Game+Phase Random effect model

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a[1]	-0.3456	0.1673	0.00518	-0.6826	-0.3529	-0.02158	1001	1000
a[2]	-0.01605	0.1586	0.01072	-0.3307	-0.01766	0.2812	1001	1000
a[3]	0.4635	0.1343	0.007974	0.2023	0.4636	0.7315	1001	1000
a[4]	0.015	0.1429	0.008007	-0.2622	0.01421	0.2884	1001	1000
a[5]	0.3023	0.133	0.007456	0.03389	0.3029	0.5663	1001	1000
a[6]	-0.0944	0.1773	0.01124	-0.4632	-0.08454	0.2192	1001	1000
a[7]	-0.2945	0.1796	0.009929	-0.6377	-0.2965	0.04007	1001	1000
a[8]	0.1076	0.139	0.007424	-0.1677	0.1148	0.3683	1001	1000
a[9]	-0.0863	0.1712	0.009767	-0.4345	-0.07274	0.2229	1001	1000
a[10]	-0.1529	0.1817	0.009951	-0.5247	-0.1521	0.2003	1001	1000
a[11]	0.1834	0.1434	0.00972	-0.1152	0.1913	0.4378	1001	1000
a[12]	-0.08201	0.1463	0.01023	-0.3712	-0.07851	0.198	1001	1000

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
d[1]	6.489E-5	0.1516	0.004588	-0.2998	0.002291	0.298	1001	1000
d[2]	-0.1544	0.1409	0.006975	-0.4213	-0.1573	0.1219	1001	1000
d[3]	-0.2192	0.1696	0.009863	-0.5517	-0.2132	0.1093	1001	1000
d[4]	-0.2428	0.1539	0.009448	-0.524	-0.2441	0.07682	1001	1000
d[5]	-0.05302	0.1426	0.008344	-0.3395	-0.04931	0.2256	1001	1000
d[6]	0.271	0.1605	0.0102	-0.06095	0.275	0.5755	1001	1000
d[7]	-0.0674	0.1754	0.01114	-0.4173	-0.06669	0.2732	1001	1000
d[8]	0.1557	0.1509	0.009445	-0.1503	0.1535	0.4543	1001	1000
d[9]	0.1019	0.1578	0.009307	-0.2103	0.1017	0.4289	1001	1000
d[10]	0.1383	0.1507	0.009702	-0.1649	0.1376	0.4261	1001	1000
d[11]	0.1211	0.1458	0.01102	-0.1612	0.1186	0.4163	1001	1000
d[12]	-0.05133	0.1538	0.007929	-0.3629	-0.04818	0.2343	1001	1000

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
z[1]	0.04585	0.09691	0.006021	-0.1019	0.02577	0.2863	1001	1000
z[2]	-0.05124	0.0874	0.009342	-0.2878	-0.03193	0.07434	1001	1000
z[3]	-0.02848	0.08015	0.00548	-0.2033	-0.02112	0.1353	1001	1000
z[4]	-0.006786	0.08619	0.003643	-0.2189	-0.001727	0.1729	1001	1000
z[5]	-0.001821	0.09594	0.003873	-0.2054	0.001233	0.2025	1001	1000
z[6]	0.04219	0.1012	0.006908	-0.1257	0.02323	0.3227	1001	1000
z[7]	-0.008067	0.08347	0.003583	-0.2061	-0.003043	0.1503	1001	1000
z[8]	0.01043	0.09337	0.002894	-0.1826	0.00528	0.2265	1001	1000
z[9]	-0.007778	0.0979	0.003479	-0.2272	-0.005232	0.1864	1001	1000
z[10]	-0.01593	0.08928	0.004968	-0.2165	-0.0064	0.1499	1001	1000
z[11]	-0.01171	0.07307	0.003218	-0.1724	-0.004245	0.1215	1001	1000
z[12]	-0.001216	0.1092	0.003197	-0.2516	-0.001124	0.2346	1001	1000

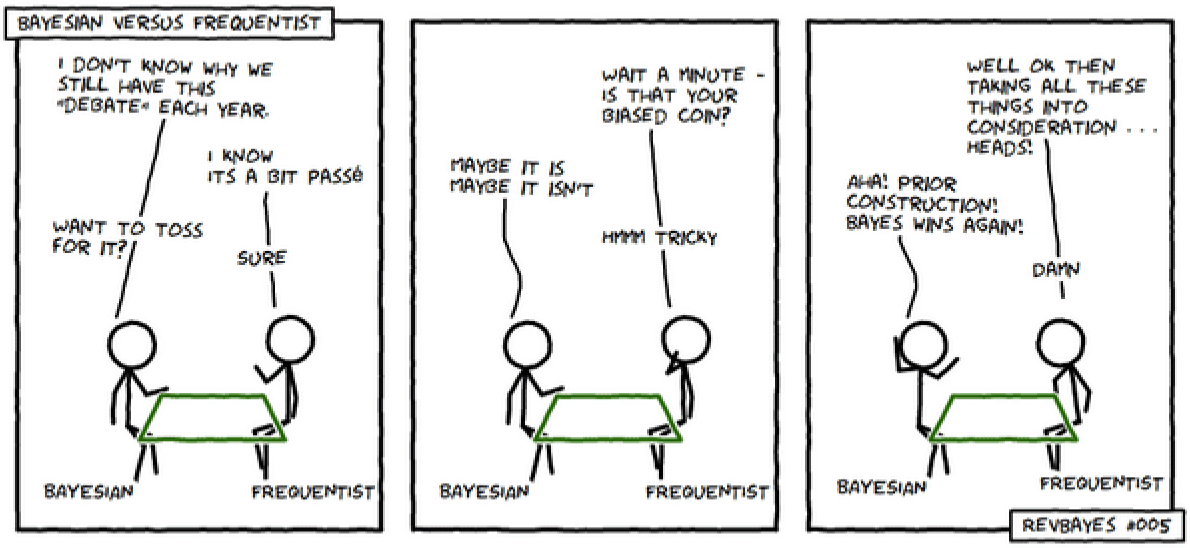
Deviance information

	Dbar	Dhat	DIC	pD
g	390.8	362.8	418.7	27.94
game	29450.0	29450.0	29450.0	1.225E-11
phase	1281.0	1281.0	1281.0	-1.202E-12
total	31120.0	31090.0	31150.0	27.94

ix. Conclusion

As we can see the best fitted model is with only game random effects instead of the fixed one or the game+phase random. This implies that the goals scored in a game by the two competing teams demonstrate a nonzero correlation.

5. BAYESIAN VS INTREDACE STATISTICS A BATTLE FOR THE AGES



Why use for our water polo data Bayesian statistics and not the normal statistics that we all know? Below is the comparison between of them two as long as the positives and negatives for each method

x. Bayesian inference

- ✓ uses probabilities for both hypotheses and data.
- ✓ depends on the prior and likelihood of observed data.
- ✓ requires one to know or construct a ‘subjective prior’.
- ✓ dominated statistical practice before the 20th century.
- ✓ may be computationally intensive due to integration over many parameters.

xi. Frequentist inference

- ✓ never uses or gives the probability of a hypothesis (no prior or posterior).
- ✓ depends on the likelihood for both observed and unobserved data.
- ✓ does not require a prior.
- ✓ dominated statistical practice during the 20th century.
- ✓ tends to be less computationally intensive.

xii. Critique of Bayesian inference

- The main critique of Bayesian inference is that a subjective prior is, well, subjective. There is no single method for choosing a prior, so different people will produce different priors and may therefore arrive at different posteriors and conclusions. Furthermore, there are philosophical objections to assigning probabilities to hypotheses, as hypotheses do not constitute outcomes of repeatable experiments in which one can measure long-term frequency. Rather, a hypothesis is either true or false, regardless of whether one knows which is the case. A coin is either fair or unfair; treatment 1 is either better or worse than treatment 2; the sun will or will not come up tomorrow.

xiii. Defense of Bayesian inference

- The probability of hypotheses is exactly what we need to make decisions.
- When the doctor tells me a screening test came back positive, I want to know what the probability is this means I'm sick. That is, I want to know the probability of the hypothesis "I'm sick".
- Using Bayes' theorem is logically rigorous. Once we have a prior all our calculations have the certainty of deductive logic.
- By trying different priors we can see how sensitive our results are to the choice of prior.
- It is easy to communicate a result framed in terms of probabilities of hypotheses.
- Even though the prior may be subjective, one can specify the assumptions used to arrive at it, which allows other people to challenge it or try other priors.
- The evidence derived from the data is independent of notions about 'data more extreme' that depend on the exact experimental.
- Data can be used as it comes in. There is no requirement that every contingency be planned for ahead of time.

xiv. Critique of frequentist inference

- It is ad-hoc and does not carry the force of deductive logic. Notions like 'data more extreme' are not well defined. The p-value depends on the exact experimental setup.
- Experiments must be fully specified ahead of time. This can lead to paradoxical seeming results. The p-value and significance level are notoriously prone to misinterpretation. Careful statisticians know that a significance level of 0.05 means the probability of a type I error is 5%. That is, if the null hypothesis is true then 5% of the time it will be

rejected due to randomness. Many (most) other people think a p-value of 0.05 means that the probability of the null hypothesis is 5%. Strictly speaking you could argue that this is not a critique of frequentist inference but, rather, a critique of popular ignorance. Still, the subtlety of the ideas certainly contributes to the problem.

xv. Defense of frequentist inference

- It is objective: all statisticians will agree on the p-value. Any individual can then decide if the p-value warrants rejecting the null hypothesis. Comparison of frequentist and Bayesian inference. Hypothesis testing using frequentist significance testing is applied in the statistical analysis of scientific investigations, evaluating the strength of evidence against a null hypothesis with data.
- The interpretation of the results is left to the user of the tests. Different users may apply different significance levels for determining statistical significance. Frequentist statistics does not pretend to provide a way to choose the significance level; rather it explicitly describes the trade-off between type I and type II errors.
- Frequentist experimental design demands a careful description of the experiment and methods of analysis before starting. This helps control for experimenter bias.
- The frequentist approach has been used for over 100 years and we have seen tremendous scientific progress.

Conclusion: each method has her own advantages and disadvantage, none is better than other just each method is better suited for different situations

6. APENDIX

MODEL#1

```
model{
                    for (i in 1:n){
                         # stochastic component
                         goals1[i]~dpois(lambda1[i])
                         goals2[i]~dpois(lambda2[i])
                         # link and linear predictor
                                log(lambda1[i])<-  mu  + a[ ht[i] ] + d[ at[i] ]
                                log(lambda2[i])<-  mu           + a[ at[i] ] + d[ ht[i] ]
                    }
      # STZ constraints
                    a[1]<-  -sum( a[2:12] )
                    d[1]<-  -sum( d[2:12] )
                    #
                    # prior distributions
                    mu~dnorm(0,0.001)
                    for (i in 2:K){
                                        a[i]~dnorm(0,0.01)
                                        d[i]~dnorm(0,0.01)
                    }

}

INITS
list( mu=0.1, a=c(NA, 0,0,0,0,0, 0,0,0,0,0, 0) , d=c(NA, 0,0,0,0,0, 0,0,0,0,0, 0)  )

DATA - LIST FORMAT
list(n=44, K=12,
ht = c(1, 2,
          6, 9, 7, 4, 2, 7, 12, 10, 11, 5, 6, 8, 9, 1, 3, 4, 1, 10, 7, 12, 2, 9,
          3, 6, 11, 5, 8, 4, 10, 9, 8, 5, 11, 3, 1, 12, 2, 3, 1, 8, 2, 5),
at = c(11, 10, 8, 5, 12, 3, 9, 1, 8, 3, 6, 4, 7, 11, 10, 12, 5, 2, 6, 5,
          8, 11, 3, 4, 9, 12, 7, 2, 1, 10, 6, 7, 4, 12, 2, 1, 8, 11, 5, 4, 12,
          11, 4, 3),
goals1=c(4, 9, 7, 7, 5, 7, 8, 3, 8, 3, 11, 7, 4, 4,
          8, 3, 7, 7, 7, 4, 7, 7, 3, 5, 13, 6, 8, 9, 6, 9, 8, 6, 6, 7, 6, 15, 7,
          5, 7, 7, 5, 14, 6, 12),
goals2=c(9, 5, 10, 9, 7, 7, 3, 5, 9, 11, 8,
          6, 4, 6, 7, 3, 6, 6, 8, 9, 8, 10, 8, 6, 6, 11, 10, 6, 7, 7, 7, 6, 7,
          7, 7, 4, 8, 8, 10, 5, 8, 12, 7, 15))
```

Model#1#1

```
model{
                for (i in 1:n){
                    # stochastic component
                    goals1[i]~dnegbin( p1[i], r)
                    goals2[i]~dnegbin( p2[i], r)
                    # link and linear predictor
                    p1[i] <- r/(r+lambda1[i])
                    p2[i] <- r/(r+lambda2[i])
                    log(lambda1[i] ) <- mu+ a[ ht[i] ] + d[ at[i] ]
                         log(lambda2[i])  <-  mu   + a[ at[i] ] + d[ ht[i] ]
                }
     # STZ constraints
                a[1]<-  -sum( a[2:12] )
                d[1]<-  -sum( d[2:12] )
                #
                # prior distrib
                r ~ dgamma( 0.001, 0.001 )
                mu~dnorm(0,0.001)
                home~dnorm(0,0.001)
                for (i in 2:K){
                                a[i]~dnorm(0,0.01)
                                d[i]~dnorm(0,0.01)
                }

}


INITS
list( r=1,mu=0.1, a=c(NA, 0,0,0,0,0, 0,0,0,0,0, 0) , d=c(NA, 0,0,0,0,0, 0,0,0,0,0, 0)  )

DATA - LIST FORMAT
list(n=44, K=12,
ht = c(1, 2,
        6, 9, 7, 4, 2, 7, 12, 10, 11, 5, 6, 8, 9, 1, 3, 4, 1, 10, 7, 12, 2, 9,
        3, 6, 11, 5, 8, 4, 10, 9, 8, 5, 11, 3, 1, 12, 2, 3, 1, 8, 2, 5),
at = c(11, 10, 8, 5, 12, 3, 9, 1, 8, 3, 6, 4, 7, 11, 10, 12, 5, 2, 6, 5,
        8, 11, 3, 4, 9, 12, 7, 2, 1, 10, 6, 7, 4, 12, 2, 1, 8, 11, 5, 4, 12,
        11, 4, 3),
goals1=c(4, 9, 7, 7, 5, 7, 8, 3, 8, 3, 11, 7, 4, 4,
        8, 3, 7, 7, 7, 4, 7, 7, 3, 5, 13, 6, 8, 9, 6, 9, 8, 6, 6, 7, 6, 15, 7,
        5, 7, 7, 5, 14, 6, 12),
goals2=c(9, 5, 10, 9, 7, 7, 3, 5, 9, 11, 8,
        6, 4, 6, 7, 3, 6, 6, 8, 9, 8, 10, 8, 6, 6, 11, 10, 6, 7, 7, 7, 6, 7,
        7, 7, 4, 8, 8, 10, 5, 8, 12, 7, 15))
```

MODEL #2+3

```
model{
                for (i in 1:n){
                    # stochastic component
                    goals1[i]~dpois(lambda1[i])
                    goals2[i]~dpois(lambda2[i])
                    # link and linear predictor
                         log(lambda1[i])<-  mu  + a[ ht[i] ] + d[ at[i] ]
                         log(lambda2[i])<-  mu   + a[ at[i] ] + d[ ht[i] ]
                }
     # STZ constraints
                a[1]<-  -sum( a[2:12] )
                d[1]<-  -sum( d[2:12] )
                #
                # prior distributions
                mu~dnorm(0,0.001)
                home~dnorm(0,0.001)
                for (i in 2:K){
                               a[i]~dnorm(0,0.01)
                               d[i]~dnorm(0,0.01)
                }

                # calculation of the predicted differences
                pred.diff[1] <- goals1[39]-goals2[39]
                pred.diff[2] <- goals1[40]-goals2[40]
     #
                # calculation of the probability of each game outcome (win/draw/loss)
                for (i in 1:2){
                   outcome[i,1] <- 1 - step( -pred.diff[i] )    # home wins
                   outcome[i,2] <- equals( pred.diff[i] , 0.0 ) # draw
                   outcome[i,3] <- 1-step( pred.diff[i] )       # home loses
                }
     #
                # calculation of the probability of each difference
                for (i in 1:2){
        pred.diff.counts[i,1] <- 1-step( pred.diff[i] + 5 )
                         for (k in 2:12){
                                        pred.diff.counts[i,k] <- equals( pred.diff[i] , k-7 ) #
                         }
        pred.diff.counts[i,13] <- step( pred.diff[i]  - 6 )
                }

}
```

INITS

```
list( mu=0.1, a=c(NA, 0,0,0,0,0, 0,0,0,0,0, 0) , d=c(NA, 0,0,0,0,0, 0,0,0,0,0, 0)  )
```

DATA - LIST FORMAT

```
list(n=40, K=12,
ht = c(1, 2,
        6, 9, 7, 4, 2, 7, 12, 10, 11, 5, 6, 8, 9, 1, 3, 4, 1, 10, 7, 12, 2, 9,
        3, 6, 11, 5, 8, 4, 10, 9, 8, 5, 11, 3, 1, 12, 2, 3),
at = c(11, 10, 8, 5, 12, 3, 9, 1, 8, 3, 6, 4, 7, 11, 10, 12, 5, 2, 6, 5,
        8, 11, 3, 4, 9, 12, 7, 2, 1, 10, 6, 7, 4, 12, 2, 1, 8, 11, 5, 4),
goals1=c(4, 9, 7, 7, 5, 7, 8, 3, 8, 3, 11, 7, 4, 4,
        8, 3, 7, 7, 7, 4, 7, 7, 3, 5, 13, 6, 8, 9, 6, 9, 8, 6, 6, 7, 6, 15, 7,
        5, NA,NA),
```

```
goals2=c(9, 5, 10, 9, 7, 7, 3, 5, 9, 11, 8,
        6, 4, 6, 7, 3, 6, 6, 8, 9, 8, 10, 8, 6, 6, 11, 10, 6, 7, 7, 7, 6, 7,
        7, 7, 4, 8, 8, NA,NA))
```

For finals

```
list(n=44, K=12,
ht = c(1, 2,
        6, 9, 7, 4, 2, 7, 12, 10, 11, 5, 6, 8, 9, 1, 3, 4, 1, 10, 7, 12, 2, 9,
        3, 6, 11, 5, 8, 4, 10, 9, 8, 5, 11, 3, 1, 12, 2, 3, 1, 8, 2, 5),
at = c(11, 10, 8, 5, 12, 3, 9, 1, 8, 3, 6, 4, 7, 11, 10, 12, 5, 2, 6, 5,
        8, 11, 3, 4, 9, 12, 7, 2, 1, 10, 6, 7, 4, 12, 2, 1, 8, 11, 5, 4, 12,
        11, 4, 3),
goals1=c(4, 9, 7, 7, 5, 7, 8, 3, 8, 3, 11, 7, 4, 4,
        8, 3, 7, 7, 7, 4, 7, 7, 3, 5, 13, 6, 8, 9, 6, 9, 8, 6, 6, 7, 6, 15, 7,
        5, 7, 7, 5, 14, 6, 12,NA),
goals2=c(9, 5, 10, 9, 7, 7, 3, 5, 9, 11, 8,
        6, 4, 6, 7, 3, 6, 6, 8, 9, 8, 10, 8, 6, 6, 11, 10, 6, 7, 7, 7, 6, 7,
        7, 7, 4, 8, 8, 10, 5, 8, 12, 7, 15,,NA))
```

MODEL #4

```
model{
                    for (i in 1:n){
                          # stochastic component
                          goals1[i]~dpois(lambda1[i])
                          goals2[i]~dpois(lambda2[i])
                          # link and linear predictor
                                 log(lambda1[i])<-  mu  + a[ ht[i] ] + d[ at[i] ]
                                 log(lambda2[i])<-  mu  + a[ at[i] ] + d[ ht[i] ]
                    }
                    # STZ constraints
                    a[1]<-  -sum( a[2:12] )
                    d[1]<-  -sum( d[2:12] )
                    # prior distributions
                    mu~dnorm(0,0.001)
                    home~dnorm(0,0.001)
                    for (i in 2:K){
                                        a[i]~dnorm(0,0.01)
                                        d[i]~dnorm(0,0.01)
                    }


      for (i in 1:K){ for (j in 1:K){
                          # replicated simulated league
                          goals1.rep[i,j]~dpois(lambda1.rep[i,j])
                          goals2.rep[i,j]~dpois(lambda2.rep[i,j])
                          # link and linear predictor
                                 log(lambda1.rep[i,j])<-  mu + a[ i ] + d[ j ]
                                 log(lambda2.rep[i,j])<-  mu           + a[ j ] + d[ i ]
                                 goal.diff.rep[i,j] <- goals1.rep[i,j]-goals2.rep[i,j]
                                 points1[i,j] <- 3*(1-step(-goal.diff.rep[i,j])) + 1*equals(goal.diff.rep[i,j],0)
                                 points2[i,j] <- 3*(1-step( goal.diff.rep[i,j])) + 1*equals(goal.diff.rep[i,j],0)
      }}
      # calculation of the total points for each team
      for (i in 1:K){
          total.points[i] <- sum( points1[i,1:12] ) -  points1[i,i] + sum( points2[1:12,i] ) -  points2[i,i]
      }
     #
     # ranking probabilities
            for (i in 1:K){
              ranks[i] <- 13-rank(total.points[], i)
              for (j in 1:K){
                  rank.probs[i,j] <- equals( ranks[i], j )
              }
            }
}

INITS
list( mu=0.1, a=c(NA, 0,0,0,0,0, 0,0,0,0,0, 0) , d=c(NA, 0,0,0,0,0, 0,0,0,0,0, 0)  )

DATA - LIST FORMAT
list(n=44, K=12,
ht = c(1, 2,
          6, 9, 7, 4, 2, 7, 12, 10, 11, 5, 6, 8, 9, 1, 3, 4, 1, 10, 7, 12, 2, 9,
          3, 6, 11, 5, 8, 4, 10, 9, 8, 5, 11, 3, 1, 12, 2, 3, 1, 8, 2, 5),
at = c(11, 10, 8, 5, 12, 3, 9, 1, 8, 3, 6, 4, 7, 11, 10, 12, 5, 2, 6, 5,
          8, 11, 3, 4, 9, 12, 7, 2, 1, 10, 6, 7, 4, 12, 2, 1, 8, 11, 5, 4, 12,
```

```
        11, 4, 3),
goals1=c(4, 9, 7, 7, 5, 7, 8, 3, 8, 3, 11, 7, 4, 4,
        8, 3, 7, 7, 7, 4, 7, 7, 3, 5, 13, 6, 8, 9, 6, 9, 8, 6, 6, 7, 6, 15, 7,
        5, 7, 7, 5, 14, 6, 12),
goals2=c(9, 5, 10, 9, 7, 7, 3, 5, 9, 11, 8,
        6, 4, 6, 7, 3, 6, 6, 8, 9, 8, 10, 8, 6, 6, 11, 10, 6, 7, 7, 7, 6, 7,
        7, 7, 4, 8, 8, 10, 5, 8, 12, 7, 15))
```

Model#fixed

```
model{
        for (i in 1:n){
                        g[i,1] <- goals1[i]
                        g[i,2] <- goals2[i]

                        g[i,1]~dpois(  lambda[i,1]  )
                        g[i,2]~dpois(  lambda[i,2]  )

                        log( lambda[i,1] ) <- mu + a[ ht[i] ] + d[ at[i] ]
                        log( lambda[i,2] ) <- mu + a[ at[i] ] + d[ ht[i] ]

                        res2[i,1] <- pow( g[i,1] - lambda[i,1], 2)
                        res2[i,2] <- pow( g[i,2] - lambda[i,2], 2)

        }

        mu~dnorm( 0, 0.001)

        a[1]  <-  -sum(a[2:K])
        d[1] <-  -sum(d[2:K])
        for (k in 2:K){
                        a[k]~dnorm( 0, 0.001)
                        d[k]~dnorm( 0, 0.001)
                        }

        ss.res <- sum(res2[1:n,1:2])
        ss.y   <- pow( sd(g[1:n,1:2]), 2)*(2*n-1)

        R2pois <- 1 - ss.res/ss.y

        for(i in 1:n){
                game[i]~dnorm(0,1)
                phase[i]~dnorm(0,1)
                }
}



INITS
list( mu=0.0,a=c(NA, 0,0,0,0,0, 0,0,0,0,0, 0),d=c(NA, 0,0,0,0,0, 0,0,0,0,0, 0)  )

DATA - LIST FORMAT
list(n=44, K=12,
```

```
ht = c(1, 2,
        6, 9, 7, 4, 2, 7, 12, 10, 11, 5, 6, 8, 9, 1, 3, 4, 1, 10, 7, 12, 2, 9,
        3, 6, 11, 5, 8, 4, 10, 9, 8, 5, 11, 3, 1, 12, 2, 3, 1, 8, 2, 5),
at= c(11, 10, 8, 5, 12, 3, 9, 1, 8, 3, 6, 4, 7, 11, 10, 12, 5, 2, 6, 5,
        8, 11, 3, 4, 9, 12, 7, 2, 1, 10, 6, 7, 4, 12, 2, 1, 8, 11, 5, 4, 12,
        11, 4, 3),
goals1=c(4, 9, 7, 7, 5, 7, 8, 3, 8, 3, 11, 7, 4, 4,
        8, 3, 7, 7, 7, 4, 7, 7, 3, 5, 13, 6, 8, 9, 6, 9, 8, 6, 6, 7, 6, 15, 7,
        5, 7, 7, 5, 14, 6, 12),
goals2=c(9, 5, 10, 9, 7, 7, 3, 5, 9, 11, 8,
        6, 4, 6, 7, 3, 6, 6, 8, 9, 8, 10, 8, 6, 6, 11, 10, 6, 7, 7, 7, 6, 7,
        7, 7, 4, 8, 8, 10, 5, 8, 12, 7, 15),
phase = c(3, 2, 3, 2, 3, 2,
        2, 3, 3, 2, 3, 2, 3, 3, 2, 3, 2, 2, 3, 2, 3, 3, 2, 2, 2, 3, 3, 2, 3,
        2, 4, 9, 11, 11, 11, 11, 7, 7, 10, 10, 8, 6, 5, 1),
game = c(1, 2,
        3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
        22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38,
        39, 40, 41, 42, 43, 44)
)
```

Model#random game

```
model{
	for (i in 1:n){
			g[i,1] <- goals1[i]
			g[i,2] <- goals2[i]

			g[i,1]~dpois(  lambda[i,1]  )
			g[i,2]~dpois(  lambda[i,2]  )

			log( lambda[i,1] ) <- mu + a[ ht[i] ] + d[ at[i] ]+gam[i]
			log( lambda[i,2] ) <- mu + a[ at[i] ] + d[ ht[i] ]+gam[i]

			res2[i,1] <- pow( g[i,1] - lambda[i,1], 2)
			res2[i,2] <- pow( g[i,2] - lambda[i,2], 2)

			gam[i] ~ dnorm( 0, tau)

	}

	mu~dnorm( 0, 0.001)

a[1]  <-  -sum(a[2:K])
	d[1] <-  -sum(d[2:K])
	for (k in 2:K){
			a[k]~dnorm( 0, 0.001)
			d[k]~dnorm( 0, 0.001)
			}

	tau~dgamma( 0.001, 0.001)


	for(i in 1:n){
		game[i]~dnorm(0,1)
		phase[i]~dnorm(0,1)
		}

}



NITS
list( mu=0.0,a=c(NA, 0,0,0,0,0, 0,0,0,0,0, 0),d=c(NA, 0,0,0,0,0, 0,0,0,0,0, 0),tau=1  )

DATA - LIST FORMAT
list(n=44, K=12,
ht = c(1, 2,
	6, 9, 7, 4, 2, 7, 12, 10, 11, 5, 6, 8, 9, 1, 3, 4, 1, 10, 7, 12, 2, 9,
	3, 6, 11, 5, 8, 4, 10, 9, 8, 5, 11, 3, 1, 12, 2, 3, 1, 8, 2, 5),
at= c(11, 10, 8, 5, 12, 3, 9, 1, 8, 3, 6, 4, 7, 11, 10, 12, 5, 2, 6, 5,
	8, 11, 3, 4, 9, 12, 7, 2, 1, 10, 6, 7, 4, 12, 2, 1, 8, 11, 5, 4, 12,
	11, 4, 3),
goals1=c(4, 9, 7, 7, 5, 7, 8, 3, 8, 3, 11, 7, 4, 4,
	8, 3, 7, 7, 7, 4, 7, 7, 3, 5, 13, 6, 8, 9, 6, 9, 8, 6, 6, 7, 6, 15, 7,
	5, 7, 7, 5, 14, 6, 12),
goals2=c(9, 5, 10, 9, 7, 7, 3, 5, 9, 11, 8,
	6, 4, 6, 7, 3, 6, 6, 8, 9, 8, 10, 8, 6, 6, 11, 10, 6, 7, 7, 7, 6, 7,
	7, 7, 4, 8, 8, 10, 5, 8, 12, 7, 15),
phase = c(3, 2, 3, 2, 3, 2,
```

```
           2, 3, 3, 2, 3, 2, 3, 3, 2, 3, 2, 2, 3, 2, 3, 3, 2, 2, 2, 3, 3, 2, 3,
           2, 4, 9, 11, 11, 11, 11, 7, 7, 10, 10, 8, 6, 5, 1),
game = c(1, 2,
           3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
           22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38,
           39, 40, 41, 42, 43, 44)
)
```

Model#random game+phase

```
model{
	for (i in 1:n){
			g[i,1] <- goals1[i]
			g[i,2] <- goals2[i]

			g[i,1]~dpois(  lambda[i,1]  )
			g[i,2]~dpois(  lambda[i,2]  )

			log( lambda[i,1] ) <- mu + a[ ht[i] ] + d[ at[i] ]+gam[i]+z[phase[i]]
			log( lambda[i,2] ) <- mu + a[ at[i] ] + d[ ht[i] ]+gam[i]+z[phase[i]]

			res2[i,1] <- pow( g[i,1] - lambda[i,1], 2)
			res2[i,2] <- pow( g[i,2] - lambda[i,2], 2)

			gam[i] ~ dnorm( 0, tau[1])

	}
			for (k in 1:12){
				z[k] ~ dnorm( 0, tau[2])
			}

	mu~dnorm( 0, 0.001)

	a[1]  <-  -sum(a[2:K])
	d[1] <-  -sum(d[2:K])
	for (k in 2:K){
			a[k]~dnorm( 0, 0.001)
			d[k]~dnorm( 0, 0.001)
			}

	tau[1]~dgamma( 0.001, 0.001)
	tau[2]~dgamma( 0.001, 0.001)


	for(i in 1:n){
		game[i]~dnorm(0,1)
		phase[i]~dnorm(0,1)
		}
}



INITS
list( mu=0.0,a=c(NA, 0,0,0,0,0, 0,0,0,0,0, 0),d=c(NA, 0,0,0,0,0, 0,0,0,0,0, 0),tau=c(1,1)  )

DATA - LIST FORMAT
list(n=44, K=12,
ht = c(1, 2,
	6, 9, 7, 4, 2, 7, 12, 10, 11, 5, 6, 8, 9, 1, 3, 4, 1, 10, 7, 12, 2, 9,
	3, 6, 11, 5, 8, 4, 10, 9, 8, 5, 11, 3, 1, 12, 2, 3, 1, 8, 2, 5),
at= c(11, 10, 8, 5, 12, 3, 9, 1, 8, 3, 6, 4, 7, 11, 10, 12, 5, 2, 6, 5,
	8, 11, 3, 4, 9, 12, 7, 2, 1, 10, 6, 7, 4, 12, 2, 1, 8, 11, 5, 4, 12,
	11, 4, 3),
goals1=c(4, 9, 7, 7, 5, 7, 8, 3, 8, 3, 11, 7, 4, 4,
	8, 3, 7, 7, 7, 4, 7, 7, 3, 5, 13, 6, 8, 9, 6, 9, 8, 6, 6, 7, 6, 15, 7,
	5, 7, 7, 5, 14, 6, 12),
goals2=c(9, 5, 10, 9, 7, 7, 3, 5, 9, 11, 8,
```

```
            6, 4, 6, 7, 3, 6, 6, 8, 9, 8, 10, 8, 6, 6, 11, 10, 6, 7, 7, 7, 6, 7,
            7, 7, 4, 8, 8, 10, 5, 8, 12, 7, 15),
phase = c(3, 2, 3, 2, 3, 2,
            2, 3, 3, 2, 3, 2, 3, 3, 2, 3, 2, 2, 3, 2, 3, 3, 2, 2, 2, 3, 3, 2, 3,
            2, 4, 9, 11, 11, 11, 11, 7, 7, 10, 10, 8, 6, 5, 1),
game = c(1, 2,
            3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
            22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38,
            39, 40, 41, 42, 43, 44)

)
```