Predicting Premier League Final Points and Rank Using Linear Modeling Techniques

By Junyuan Gao

Advisor: David Aldous

Index

I. Introduction

Soccer, one of the most popular sports in the world, has its own fascination. Soccer fans enchant in the tense and exciting moments of goals, especially those last-gasp goals that determine the game result and then determine the final rank and points of teams. For example, in the 2015-16 season of English Premier League, the dark horse team Leicester, who just ascended to the Premier League in 2014-15 season, surprisingly beat all the other teams and won the champion. This phenomenon reflects one of the most charming part of soccer— complexity, which makes the game result hard to be predicted.

Though soccer game results and team ranks are hard to predict, curious people always want to figure out the keys to determine the game result for their own reasons. Betting companies has to correctly, or as correctly as possible, predict the game results and ranks since they have to design a series of odds that produce stable profit from gamblers. Their methodology might be complex and various—from analyzing the strength of two teams and the possible strategies of two coachs, to the choice of the referee at the game day, the injury situation of two teams and both teams' future schedule, etc. Gamblers and sports fans want to predict the game results and ranks correctly since gamblers want to predict correctly since gamblers want to earn money from betting companies and gain pleasure from correctly predict their favorite club winning the game and the seasonal championship. Restricted to the lack of information and experience in the industry, they have to make prediction based on less parameters such as general performance of two teams in this season(which can be easily obtained from game table), historical game records and odds from betting companies.

As a statistician and a soccer fan, I mainly focus on predicting the game results using statistical modeling techniques. I choose to predict the game results and the team ranks in a very straight way—predicting the number of the goals for each team. The reason I choose to predict the number of goals is that regardless what strategies that coachs use or what types of the goals are, whenever a team achieve more goals than the other, that team will win the game. Within each game result produced, I can easily generate the results to make a final table contains team ranks and final points. The goal of the project is to predict the final ranks and points of 2016-17 premier league season. This goal will be achieved mainly in following steps:

(1) Data collection and re-organization in order to be used to construct prediction models
(2) Several models will be evaluated and tested on 2015-16 season
(3) Select the one among those models
(4) Predictions will be made via the best model on selected in step(2)

II. Description of Data

The data that I used in this project are collected from github and www.premierleague.com. I collected the detailed match data of 2015-16 premier league season, which can be regard as my training set, from github and obtained the completed game results and future game schedule of 2016-17 premier league season, which can be regard as the prediction set, from www.premierleague.com. Moreover, all execution on data is under R.

The original data is a data frame with 380 rows and 49 columns, in which each row stands for full information of one game in 2015-16 season. The information contains team names, managers of teams, general information(date, location, name of referee, etc.) and detailed statistics(e.g. number of assists, number of goals, number of tackles, etc.). We can have a rough glance of the data in figure 1:

	assists_away_team	assists_home_team	attendance	away_goals	away_goals_details	away_manager	away_team	blocks_away_team
1	0	3	74363	1	Chris Smalling (90+3 OG)	Eddie Howe	Bournemouth	
2	0	3	60007	0		Eric Black	Aston Villa	
3	1	0	41494	1	Daniel Drinkwater (82)	Claudio Ranieri	Leicester	
4	0	0	36691	0		Alex Neil	Norwich	
5	1	4	52183	1	Erik Lamela (60)	Mauricio Pochettino	Spurs	
6	0	2	31313	1	Jason Puncheon (64)	Alan Pardew	Crystal Palace	
7	1	1	27721	1	Michail Antonio (23)	Slaven Bilic	West Ham	
8	0	0	20934	1	Kelechi Iheanacho (5)	Manuel Pellegrini	Man City	
9	2	1	21012	2	Jack Rodwell (39),Jeremain Lens (51)	Sam Allardyce	Sunderland	
10	1	1	26196	1	Jordon Ibe (23)	Jürgen Klopp	Liverpool	
11	1	0	43210	1	Eden Hazard (32)	Guus Hiddink	Chelsea	
12	1	2	26279	2	Troy Deeney (11),Odion Ighalo (51)	Enrique Sánchez Flores	Watford	

Figure 1: first 12 rows and 8 columns of original data frame

However, the data is too messy for the further research—as I mentioned in part I, what I need is only the number of goals of home and away teams, so I simplify the original data and produced the reduced data that only contains 380 rows and 4 columns, which can be seen in Figure 2.

Filter

	home_team	away_team	home_goals	away_goals
1	Man Utd	Bournemouth	3	1
2	Arsenal	Aston Villa	4	0
3	Chelsea	Leicester	1	1
4	Everton	Norwich	3	0
5	Newcastle	Spurs	5	1
6	Southampton	Crystal Palace	4	1
7	Stoke	West Ham	2	1
8	Swansea	Man City	1	1
9	Watford	Sunderland	2	2
10	West Brom	Liverpool	1	1
11	Liverpool	Chelsea	1	1
12	Norwich	Watford	4	2
13	Sunderland	Everton	3	0

Figure 2: first 13 rows and 4 columns of reduced data.

After obtaining the original match results in 2015-16 season, I created a function Table() to generate the match results into a result table. Besides the similar statistics on the other result tables, this table contains detailed statistics like the numbers of wins that particular team play as home team and so on. The 2015-16 season table is shown

on Figure 3.

	Team	P	HW	HD	HL	HF	HA	AW	AD	AL	AF	AA	Hpts	Apts	GD	Points
1	Leicester	38	12	6	1	35	18	11	6	2	33	18	42	39	32	81
2	Arsenal	38	12	4	3	31	11	8	7	4	34	25	40	31	29	71
3	Spurs	38	10	6	3	35	15	9	7	3	34	20	36	34	34	70
4	Man City	38	12	2	5	47	21	7	7	5	24	20	38	28	30	66
5	Man Utd	38	12	5	2	27	9	7	4	8	22	26	41	25	14	66
6	Southampton	38	11	3	5	39	22	7	6	6	20	19	36	27	18	63
7	West Ham	38	9	7	3	34	26	7	7	5	31	25	34	28	14	62
8	Liverpool	38	8	8	3	33	22	8	4	7	30	28	32	28	13	60
9	Stoke	38	8	4	7	22	24	6	5	8	19	31	28	23	-14	51
10	Chelsea	38	5	9	5	32	30	7	5	7	27	23	24	26	6	50
11	Everton	38	6	5	8	35	30	5	9	5	24	25	23	24	4	47
12	Swansea	38	8	6	5	20	20	4	5	10	22	32	30	17	-10	47
13	Watford	38	6	6	7	20	19	6	3	10	20	31	24	21	-10	45
14	West Brom	38	6	5	8	20	26	4	8	7	14	22	23	20	-14	43
15	Crystal Palace	38	6	3	10	19	23	5	6	8	20	28	21	21	-12	42
16	Bournemouth	38	5	5	9	23	34	6	4	9	22	33	20	22	-22	42
17	Sunderland	38	6	6	7	23	20	3	6	10	25	42	24	15	-14	39
18	Newcastle	38	7	7	5	32	24	2	3	14	12	41	28	9	-21	37
19	Norwich	38	6	5	8	26	30	3	2	14	13	37	23	11	-28	34
20	Aston Villa	38	2	5	12	14	35	1	3	15	13	41	11	6	-49	17

Figure 3: The detailed result table of 2015-16 Premier League season. P=number of matches played, H= Home, A= Away, w=number of winning games, D=number of draw games, L=number of losing games, HF/AF=Goal scored in home/away games, HA/AA= number of goals against in home/away games, Hpts/Apts=points earned in home/away games, GD= goal difference.

By the same way, game results data and result table of 2016-17 season can also be generated. Within this data, we are ready to step to the next part.

III. Preliminary Analysis

At first, I tried to figure out whether there is a trend, regardless of the difference between home game and away game, of the number of goals in each match. The following visualization can provide a directly perception:

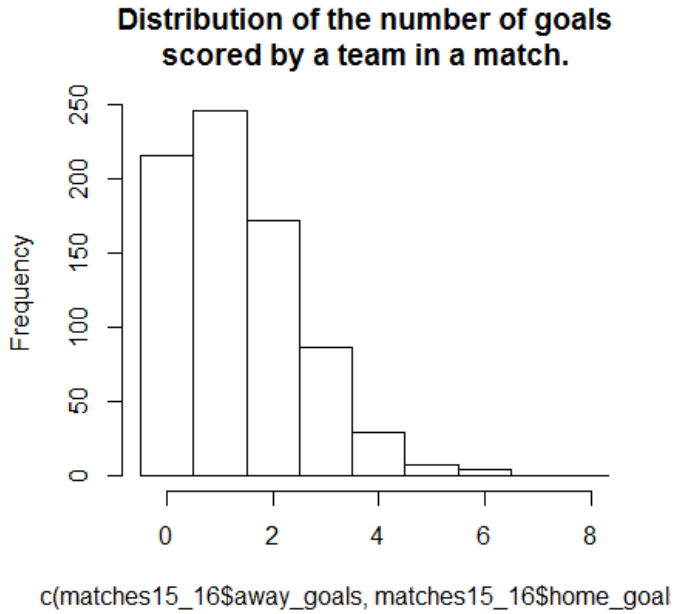

Figure 4: Distribution of numbers of goals of a team in one match (regardless the difference of home or away)

Inspired by Rasmus Baath's paper *"Modeling Match Results in Soccer using a Hierarchical Bayesian Poisson Model"*, I realized that if we assume both teams have equal probability of making a goal in each chance and both teams have many chances in the equally long game time(90 minutes), the distribution of the number of goals should follows a Poisson Distribution. To intuitively confirm this hypothesis, I plotted a random draw of a Poisson Distribution whose mean is equal to the mean of number of goals in each match and resulted in the following graph:

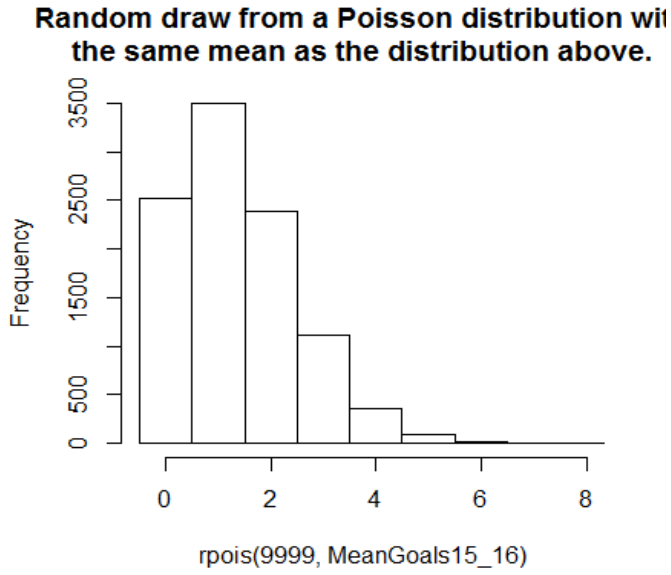

Figure 5: Distribution of results of random draw from Poisson(mean= the mean of goals scored in one match in 2015-16 season).

Observing Figure 4 and 5, we can easily detect that they looks really same, which strengthen the power of this hypothesis. However, to officially check whether it's true or not, I made a hypothesis testing by using function goodfit() from package "vcd". I set the null hypothesis *H_0: The distribution of number of goals in one game is approximately Poisson distributed*. The test result can be viewed through figure 6:

```
> pois_fit_1516 <-  goodfit(matches15_16$away_goals+matches15_16$home_goals,
+ type= "poisson", method= "ML")
> summary(pois_fit_1516)

        Goodness-of-fit test for poisson distribution

                      X^2 df  P(> X^2)
Likelihood Ratio 8.457959  8 0.3900596
```

Figure 6: Test statistics of goodness-of-fit test for Poisson distribution

This test statistics shows that $X_t^2 = 8.457959$ with 8 degrees of freedom and p-value= 0.3900596, which indicates that the probability of the data follows the distribution is approximately 39%. Though it looks a little bit small, it is quite significant since our amount of data is also large. Just as Prof. Aldous noted: All models are wrong but some are useful. Thus, by intuitively thinking, empirical observation and

statistical test, I think that the test statistics could be a strong evidence to accept my null hypothesis that the distribution of number of goals in each match in 2015-16 season follows Poisson distribution.

IV. Model Construction and Simulation

After accepting the assumption of the number of goals in each match follows Poisson distribution, I can start construct regression models based on the assumption. In this part, two models are considered:

Model 1: Poisson regression separately on 2 parameters: home goals(Offence) and away goals(Defence) of teams.
Model 2: Based on Model 1, consider an extra parameter—home advantage parameter.

Theoretically, the Poisson regression formula for this project can be represented as $Y=\exp(X\beta)$, where Y is a vector of dependent variable that consists of the home goals and away goals in games, X is a matrix of explanatory variables that records the home and away teams corresponding to the games. β is a vector containing the parameters, Offence and Deffence, of the model. Note that Y and β are at length 2n which is 2 times of number of mathces since each of 20 teams has its Offence parameter and Defence parameter, and each of the times will appear as either a home team or an away team. Thus we can say Y and β are of the forms: $Y=(y_{i1,\,j1}{}^{1},\ y_{i1,\,j1}{}^{1},\ \ldots)^{T}$ and $\beta=(O_1,\ \ldots,O_{20},\ D_1,\ \ldots,D_{20})^{T}$, where $y_{a,b}{}^{i}$ is the number of goals scored by team a versus team b in game i; O_j and D_j stands for the Offence and Defence parameter of team j. To be specific, I will show readers the structure with a naïve example:

Example: Assume we have 3 teams: Man Utd, Man City and Chelsea plays 3 games pairwisely. Then we can write following table and representations:

Table 1: example of 3 games

Game	Home team	Away team	# of home goals	# of away goals
1	Man Utd	Chelsea	0	3
2	Man City	Man Utd	2	2
3	Chelsea	Man city	3	1

Hence, we have $Y=(0, 3, 2, 2, 3, 1)^{T}$ $\beta=(O_{MU}, O_{Che}, O_{MC,} D_{MU}, D_{Che}, D_{MC,})^{T}$ and

$$X=\begin{array}{c} \begin{matrix} MU & Che & MC & MU & Che & MC \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & -1 & 0 \end{pmatrix} \end{array} \begin{matrix} \text{Game1} \\ \text{Game1} \\ \text{Game2} \\ \text{Game2} \\ \text{Game3} \\ \text{Game3} \end{matrix}$$

If we consider the home advantage parameter in model 2, our β and X will change into $\beta = (O_{MU}, O_{Che}, O_{MC,} D_{MU}, D_{Che}, D_{MC,} \delta)^T$ and

$$X = \begin{array}{c} \begin{matrix} MU & Che & MC & MU & Che & MC & home \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 \end{pmatrix} \end{array} \begin{matrix} Game1 \\ Game1 \\ Game2 \\ Game2 \\ Game3 \\ Game3 \end{matrix}$$

Moreover, to construct both model 1 and model 2 in practice, the following guideline is considered:

(1) Estimate the Poisson parameters of each team based on the average amount of home goals and away goals in each match.
(2) Produce a table of probabilities for the set of results possible for a football match based on (1)
(3) Simulate game results of a season based on (1) and (2), then using bootstrap to diminish the randomness

(1)

For the first step, I set the mean of the home/away goals of one team in the season as its Poisson parameter λ, and create a function Y_beta_x() to calculate the Poisson parameters. Inside the function, the build-in function glm() is used to calculate the Poisson parameters. However, when dealing with model 2, I observed that the matrix X in this case contains 41 columns while the rank is 40, which means that the glm function will not produce a unique least square estimator of β. So I just modified the function Y_beta_x() to make Y vector fit on X after reduced the 1st column. Equivalently, we can write the Poisson regression R code as

glm(Y~ 0+ X_reduced_1st Column, family= poisson).

Within this modification, the number of columns of X is reduced to 40 and the function will return a least square estimate of β. The result of the estimated parameters in 2015-16 season is shown on Figure 7:

```
$teams
                    Offence     Defence
Arsenal          0.000000000  0.35855066
Aston Villa     -0.664493242 -0.76853260
Bournemouth     -0.327721321 -0.75232128
Chelsea         -0.032703291 -0.64270991
Crystal Palace  -0.346872034 -0.36060233
Everton         -0.056400776 -0.64127424
Leicester        0.116919476 -0.14183498
Liverpool       -0.050237969 -0.33148921
Man City         0.130116989 -0.29694213
Man Utd         -0.204010550  0.57087503
Newcastle       -0.449563822 -0.39886925
Norwich         -0.459473387 -0.62162232
Southampton      0.008056722 -0.33511046
Spurs            0.103694138  0.04143159
Stoke           -0.359154434 -0.40262400
Sunderland      -0.412864514 -0.21802934
Swansea         -0.361387937 -0.22020536
Watford         -0.374781130 -0.16833462
West Brom       -0.332334074 -0.48385036
West Ham         0.011013540 -0.50235427

$home
[1] 0.346402
```

Figure 7: Estimated Offence and Defence parameters of each teams as well as the home advantage parameter of model 2.

(2)

For the second step, I designed a function ProbTable() that produce a probability matrix to show the probability of possible game result of two teams. The basic formula to calculate the probability of a certain outcome is:

For team A, B whose number of goals scored are j and k, P(A v.s B has result j v.s k)= P(# of goals of A= j) x P(# of goals of B= k).

For each team A, B whose number of goals scored are j and k, I used function dpois() to obtain the probability P(A=j) under the Pois(O_A- D_B) distribution and obtain the probability P(B=k) under the Pois(O_B- D_A) distribution. When using model 2, since the home parameter is added, the distributions correspondingly change into Pois(O_A- D_B+δ) and Pois(O_B- D_A). An example in figure 8 shows the probability matrix of the possible outcomes of the game (in season 2015-16) between Leicester and Manchester United:

```
> ProbTable(Y_beta_x(reduced_matches15_16), "Leicester", "Man Utd")
       0     1    2    3    4    5    6 7+
0  15.92 14.29 6.42 1.92 0.43 0.08 0.01  0
1  14.96 13.43 6.03 1.81 0.41 0.07 0.01  0
2   7.03  6.31 2.83 0.85 0.19 0.03 0.01  0
3   2.20  1.98 0.89 0.27 0.06 0.01 0.00  0
4   0.52  0.46 0.21 0.06 0.01 0.00 0.00  0
5   0.10  0.09 0.04 0.01 0.00 0.00 0.00  0
6   0.02  0.01 0.01 0.00 0.00 0.00 0.00  0
7+  0.00  0.00 0.00 0.00 0.00 0.00 0.00  0
```

Figure 8: The probability matrix of the outcomes of the game between Leicester City and Manchester United(in percentage, e.g. 0-0 has 15.92% chance). The numbers 0~7+ on the row indicates the possible number of goals of Leicester while the numbers on the column indicates the number of goals of Manchester United.

(3)

For the third step, I designed a function GameResult() to simulate the game results for a whole season based on repetitively using the methods I mentioned in part (1) and (2). After the game results of a whole season created, I used the Table() function mentioned above to generate the game results into a final result table, which is more clear and readable to readers. The following figure 9 is a nice example of the result of one-time simulation of a whole season:

```
> Table(pro_GameResult(pois_para_1516))
               Team  P HW HD HL HF HA AW AD AL AF AA Hpts Apts  GD Points
1             Spurs 38 11  3  5 43 31 12  5  2 42 24   36   41  30     77
2         Leicester 38  9  6  4 33 20 12  4  3 39 25   33   40  27     73
3          Man City 38 10  3  6 38 30 11  5  3 38 17   33   38  29     71
4           Arsenal 38  8  5  6 23 28 12  3  4 40 27   29   39   8     68
5         West Ham 38  7  6  6 26 21 11  4  4 38 27   27   37  16     64
6           Man Utd 38  9  4  6 29 20  9  5  5 26 18   31   32  17     63
7        Sunderland 38  9  4  6 29 21  9  4  6 29 22   31   31  15     62
8      Southampton 38  7  7  5 29 28 10  3  6 29 19   28   33  11     61
9        Liverpool 38  9  2  8 35 31  9  2  8 23 23   29   29   4     58
10         Chelsea 38  5  5  9 31 37  8  3  8 35 27   20   27   2     47
11     Bournemouth 38  6  4  9 16 23  6  6  7 27 22   22   24  -2     46
12         Everton 38  5  4 10 18 26  7  5  7 26 28   19   26 -10     45
13         Norwich 38  6  4  9 17 29  6  4  9 18 28   22   22 -22     44
14           Stoke 38  8  2  9 29 24  4  5 10 19 27   26   17  -3     43
15         Swansea 38  5  8  6 24 24  5  5  9 17 34   23   20 -17     43
16       Newcastle 38  6  5  8 23 26  4  6  9 20 24   23   18  -7     41
17 Crystal Palace 38  4  6  9 17 31  6  4  9 25 32   18   22 -21     40
18       West Brom 38  3  5 11 20 36  5  7  7 17 22   14   22 -21     36
19         Watford 38  8  2  9 20 27  1  7 11 14 31   26   10 -24     36
20     Aston Villa 38  4  5 10 14 27  4  3 12 18 37   17   15 -32     32
```

Figure 9: A result table of one-time simulation of games of the whole 2015-16 season of model 1

However, since one-time simulation might be badly affected by randomness, I used the 1000 times bootstrap to produce game results of 1000 seasons and then take average of them, which will weaken the effect of randomness. Since this part will mainly work for the next section, only the bootstrapping game results, average rank and final points are recorded after the bootstrap. Figure 10 is a comparison of average ranks and final points between model 1 and model 2 after bootstrapping.

```
> reverse_pro_btsp1516
             Team  Rank Points
1       Leicester  5.00  63.63
2         Arsenal  5.13  63.38
3           Spurs  4.60  64.70
4        Man City  3.69  67.14
5         Man Utd 11.24  50.80
6     Southampton  7.31  58.69
7        West Ham  5.20  63.28
8       Liverpool  6.46  60.53
9           Stoke 14.29  44.69
10        Chelsea  8.05  57.44
11        Everton  7.75  57.46
12        Swansea 14.70  44.02
13        Watford 14.60  44.07
14      West Brom 16.71  38.71
15 Crystal Palace 14.69  43.33
16    Bournemouth 12.01  48.73
17     Sunderland 11.85  49.74
18      Newcastle 12.95  47.61
19        Norwich 15.14  43.04
20    Aston Villa 18.63  32.49
```

```
> reverse_btsp1516
             Team   Rank Points
1       Leicester  3.894 70.968
2         Arsenal  2.236 78.664
3           Spurs  2.972 74.969
4        Man City  5.151 66.399
5         Man Utd  2.753 75.983
6     Southampton  7.222 60.050
7        West Ham  9.238 54.913
8       Liverpool  8.009 57.935
9           Stoke 13.430 45.183
10        Chelsea 12.267 47.777
11        Everton 12.834 46.762
12        Swansea 11.043 50.556
13        Watford 10.373 52.177
14      West Brom 14.434 43.065
15 Crystal Palace 12.714 46.892
16    Bournemouth 17.751 33.612
17     Sunderland 11.876 48.706
18      Newcastle 14.684 42.331
19        Norwich 17.571 34.003
20    Aston Villa 19.548 24.413
```

Figure 10: Comparison of average ranks and final points between model 1 and model 2 after bootstrapping. Left: model 1 Right: model 2

Moreover, I created a function Accuracy() to calculate the bias= E(λ-λ_hat) and the standard deviation dev= $\frac{n}{n-1} sqrt\bar{x^2} - \bar{x}^2$ of every parameters of each team over the bootstrap process. The smaller the bias and standard deviation are, the less randomness does the simulation have. Figure 11 is the generated result of bias and standard deviation of model 2 after 1000-times bootstrapping:

```
$teams
              Offence.bias Offence.sd Defence.bias Defence.sd
Arsenal        0.000000000  0.0000000   0.06803891  0.3196250
Aston Villa    0.079779133  0.1439219   0.08625677  0.1901382
Bournemouth    0.244091299  0.1475844   0.07803333  0.1947184
Chelsea        0.290773882  0.1509407   0.08338976  0.2045983
Crystal Palace 0.050801720  0.1374288   0.08470835  0.2311393
Everton        0.279980793  0.1473972   0.08395303  0.2018166
Leicester      0.133735037  0.1358800   0.08053174  0.2632044
Liverpool      0.154747739  0.1373569   0.06936378  0.2399200
Man City       0.187560250  0.1419385   0.07628921  0.2320731
Man Utd       -0.114217349  0.1223661   0.06046856  0.3653184
Newcastle      0.010778424  0.1431382   0.08485564  0.2355747
Norwich        0.114778857  0.1437989   0.09984997  0.2041321
Southampton    0.167437384  0.1403309   0.07196729  0.2439356
Spurs          0.087187885  0.1350762   0.07614321  0.2912141
Stoke          0.065452210  0.1414936   0.09022039  0.2340358
Sunderland    -0.025692483  0.1357623   0.08521107  0.2447434
Swansea       -0.007526085  0.1347582   0.09324415  0.2472927
Watford       -0.031944106  0.1289188   0.05755221  0.2586975
West Brom      0.107727981  0.1431044   0.07313586  0.2239881
West Ham       0.232602845  0.1515530   0.06593220  0.2196013

$home.bias
[1] -0.1331349

$home.sd
[1] 0.04092217
```

Figure 11: bias and std deviation of parameters model 2 over 1000-times bootstrapping

From the figure, we can see that the standard deviation for most of parameters are less than 0.30, which is a very small number and indicates that the set of parameters using the original ones are very closed to the original estimators I got in part (1). Furthermore, the bias values also appears to be very small, which shows that little bias occurred during the 1000-times bootstrapping process. Both the results of bias and deviation reflect that my method of estimating parameters is doing well, and surprisingly, the low bias and standard deviation of home parameter indicates that the home advantage parameter is useful.

V. Test and Model Selection

The object of this section is to select the best model based on the real world result of 2015-16 Premier League season. In order to choose the best model for the future prediction, I designed three tests to determine which model is optimal.

(1)

The first test is the non-parametric Wilcoxon rank sum test on team ranks and final points towards model 1 and model 2. Observing and comparing figure 10 and figure 3 by eye, we can see that both 2 models have a nice prediction of the top 6 rank teams and the bottom 3 rank teams towards the real-world result. However, it's hard to determine the goodness of simulation of those middle-ranked teams. By using Wilcoxon rank sum test, I can quantitatively determine the rank sum of two simulated tables and figure out its goodness of simulation. I set the null hypothesis *H_0: The simulation ranks does not shift from the real-world ranks* and test result is shown on figure 12.

```
> wilcox.test(reverse_pro_btsp1516$Rank, c(1:20), paired=TRUE)

        Wilcoxon signed rank test with continuity correction

data:  reverse_pro_btsp1516$Rank and c(1:20)
V = 104, p-value = 0.9851
alternative hypothesis: true location shift is not equal to 0

Warning message:
In wilcox.test.default(reverse_pro_btsp1516$Rank, c(1:20), paired = TRUE)
  无法精確計算带连结的p值
> wilcox.test(reverse_btsp1516$Rank, c(1:20), paired=TRUE)

        Wilcoxon signed rank test

data:  reverse_btsp1516$Rank and c(1:20)
V = 106, p-value = 0.9854
alternative hypothesis: true location shift is not equal to 0
```

Figure 12: Wilcoxon rank test statistics for model 1(above) and model 2(bottom)

Looking at the test statistics of two models, I saw that they are really same: both of them fail to reject null hypothesis due to the really large p-value, and there is nearly no difference between their rank sum and p-value. To classify the best model more

efficiently, I designed the second test.

(2)

The second test is a percentage test—that is, to test the percentage of the teams occur to be at their correct rank(i.e. the real-world rank) in the simulation. In this part, I created a function percentage_test() to solve the problem. Moreover, I calculated the sum of percentages of each team. The larger the summation is, the more accurate the model performs. Test results is shown on figure 13.

```
> percent.pro
             Team percentage
1        Leicester     18.00%
2          Arsenal     13.00%
3            Spurs      7.00%
4         Man City     15.00%
5          Man Utd      5.00%
6      Southampton      8.00%
7         West Ham     10.00%
8        Liverpool      6.00%
9            Stoke      3.00%
10         Chelsea      6.00%
11         Everton     10.00%
12         Swansea      6.00%
13         Watford     10.00%
14       West Brom      4.00%
15 Crystal Palace      9.00%
16     Bournemouth      6.00%
17      Sunderland      7.00%
18       Newcastle      3.00%
19         Norwich     12.00%
20     Aston Villa     44.00%
> sum(percent.pro$percentage)
[1] 202.00%
```

```
> percent.me
             Team percentage
1          Arsenal     25.80%
2      Aston Villa     70.90%
3      Bournemouth      7.20%
4          Chelsea      9.10%
5  Crystal Palace     10.90%
6          Everton      8.00%
7        Leicester      8.40%
8        Liverpool     14.30%
9         Man City     18.70%
10         Man Utd      8.50%
11       Newcastle     11.30%
12         Norwich     30.70%
13     Southampton     17.10%
14           Spurs     21.50%
15           Stoke      5.30%
16       Sunderland      4.60%
17          Swansea      9.80%
18          Watford      8.30%
19        West Brom     10.70%
20         West Ham     13.60%
> sum(percent.me$percentage)
[1] 314.70%
```

Figure 13: Percentage of teams at their right place and the summation of the percentages. Left: model 1 Right: Model 2

From the test results, I noticed that model 2 has much better accuracy on middle-ranked teams and has an average 5.635% higher accuracy on each team than model 1, which indicates that generally speaking model 2 performs a significantly higher accuracy than model 1.

(3)

The third test that I designed is about the accuracy of simulating the champion, which is measured by the percentage of each teams win the champion in simulated 2015-16 season. Since Leicester City was a surprising dark horse in last year, I required my best model for predicting 2016-17 season to do its best on predicting the championship without influence the accuracy of other ranks. To complete the test, I wrote a function rank1_rate() and the next figure is the result of the tests:

```
> (rank1.rate.pro <- rank1_rate(
             Team percentage
1       Leicester     18.00%
2         Arsenal     14.00%
3            Spurs     22.00%
4        Man City     23.00%
5         Man Utd      0.00%
6      Southampton      2.00%
7        West Ham      8.00%
8        Liverpool      4.00%
9            Stoke      0.00%
10         Chelsea      4.00%
11         Everton      4.00%
12         Swansea      0.00%
13         Watford      0.00%
14       West Brom      0.00%
15 Crystal Palace      0.00%
16     Bournemouth      0.00%
17      Sunderland      1.00%
18       Newcastle      0.00%
19         Norwich      0.00%
20     Aston Villa      0.00%
```

```
> (rank1.rate.me <- rank1_rate(bts
             Team percentage
1         Arsenal     40.50%
2      Aston Villa      0.00%
3      Bournemouth      0.00%
4          Chelsea      0.00%
5   Crystal Palace      0.00%
6          Everton      0.00%
7        Leicester      8.40%
8        Liverpool      0.20%
9         Man City      2.90%
10         Man Utd     24.90%
11       Newcastle      0.00%
12         Norwich      0.00%
13     Southampton      0.20%
14           Spurs     22.80%
15           Stoke      0.00%
16      Sunderland      0.00%
17         Swansea      0.00%
18         Watford      0.00%
19       West Brom      0.00%
20        West Ham      0.10%
```

Figure 14: Test results of the percentages of win a champion. Left: model 1 Right: model 2

From the above figure, I saw that model 1 does better in predicting the champion. However, an interesting appearance is that in model 2, all the top 6 teams except Leicester has extremely high percentage to win the champion while on the other side, many middle-ranked teams and relegation avoiders still have considerable chance to win the chance, which is ridiculous and might indicates that even though model 1 performs better in predicting the champion, it sacrifices a lot in accuracy of other ranks.

After general consideration of all 3 tests, I choose model 2 as my prediction model in the next section.

VI. Prediction

Knowing that model 2 is the better one, now I start predict the 2016-17 Premier League final points and team ranks. Since there were 3 teams from 2015-16 season degraded down to the Football League Championship and 3 new teams upgraded to Premier League, I can't simply take the model 2 in 2015-16 season to predict the results in 2016-17 season. Therefore, I chose to re-construct a model that based on the ideas of model 2 in 2015-16 season.

One crucial problem occurred in this part is the estimation of Poisson parameters of each team. Since the season is still in progress, I imported and select the completed 14 rounds of game results(last updated at 12/06/2016, total 14 rounds) as training sets and estimate the Offence, Defence and home parameters to predict the result of rest of games. Like what I have shown on section IV, I predicted the result of rest of games and combined them with the finished games to produce the following predicted final table of 2016-17 Premier League Season:

```
> premier1617
                   Team  P HW HD HL HF HA AW AD AL AF AA Hpts Apts  GD Points
1               Chelsea 14  6  0  1 20  4  5  1  1 12  7   18   16  21     34
2               Arsenal 14  4  2  1 15  9  5  2  0 18  5   14   17  19     31
3             Liverpool 14  5  1  0 19  4  4  2  2 16 14   16   14  17     30
4              Man City 14  3  3  1 13  8  6  0  1 17  7   12   18  15     30
5                 Spurs 14  5  2  0 14  4  2  4  1 10  6   17   10  14     27
6               Man Utd 14  2  4  1 10  6  3  2  2  9 10   10   11   3     21
7  West Bromwich Albion 14  3  2  2 13 10  2  3  2  7  7   11    9   3     20
8               Everton 14  3  4  0 10  5  2  1  4  7 11   13    7   1     20
9                 Stoke 14  3  1  3  9 11  2  3  2  7  8   10    9  -3     19
10          Bournemouth 14  4  1  2 14  9  1  2  4  5 13   13    5  -3     18
11              Watford 14  3  1  3 10 10  2  2  3  8 14   10    8  -6     18
12          Southampton 14  3  3  1  7  5  1  2  4  6 10   12    5  -2     17
13        Middlesbrough 14  2  1  4  6  7  1  5  1  7  8    7    8  -2     15
14      Crystal Palace 14  2  1  4 11 10  2  1  4 13 16    7    7  -2     14
15              Burnley 14  4  1  3 11  8  0  1  5  1 15   13    1 -11     14
16            Leicester 14  3  3  1 11  6  0  1  6  6 18   12    1  -7     13
17             West Ham 14  2  2  3  7 14  1  1  5  8 15    8    4 -14     12
18            Sunderland 14  2  1  4 10 14  1  1  5  4 10    7    4 -10     11
19                  Hull 14  2  1  4  6 12  1  1  5  5 17    7    4 -18     11
20               Swansea 14  1  2  4 10 16  1  1  5  6 15    5    4 -15      9
```

```
> Table(finalresult)
                   Team  P HW HD HL HF HA AW AD AL AF AA Hpts Apts  GD Points
1               Chelsea 38 18  0  1 58  4 15  2  2 30  9   54   47  75    101
2                 Spurs 38 17  2  0 36  4 11  7  1 25  8   53   40  49     93
3             Liverpool 38 15  3  0 48  8 12  6  2 36 21   48   42  55     90
4               Arsenal 38 15  3  1 53 15 10  5  4 34 21   48   35  51     83
5              Man City 38 12  6  1 39 15 10  5  4 34 21   42   35  37     77
6               Everton 38 10  8  1 22  9  7  5  7 14 16   38   26  11     64
7           Southampton 38 12  6  1 19  7  5  7  7 11 14   42   22   9     64
8               Man Utd 38 11  5  3 27  9  5  8  6 18 21   38   23  15     61
9         Middlesbrough 38  8  7  4 19 11  2 11  6 13 21   31   17   0     48
10 West Bromwich Albion 38  8  6  5 31 24  4  5 10 15 28   30   17  -6     47
11          Bournemouth 38 10  5  4 27 18  1  6 12  9 28   35    9 -10     44
12            Leicester 38  9  6  4 20 10  0  6 13 10 30   33    6 -10     39
13      Crystal Palace 38  7  5  7 24 21  2  6 11 21 36   26   12 -12     38
14              Burnley 38  9  5  6 16 11  0  5 13  4 28   32    5 -19     37
15              Watford 38  7  4  8 21 23  2  6 11 12 34   25   12 -24     37
16                Stoke 38  5  6  8 20 25  2  5 12 13 32   21   11 -24     32
17            Sunderland 38  4  5 10 21 34  1  2 16  9 44   17    5 -48     22
18               Swansea 38  3  7  9 21 32  1  2 16 13 49   16    5 -47     21
19             West Ham 38  3  6 10 16 35  1  2 16 14 47   15    5 -52     20
20                 Hull 38  2  8  9 12 26  1  2 16  6 42   14    5 -50     19
>
```

Figure 15: Top: Result table of finished 14 rounds of games(up to 12/06/2016)

Bottom: Predicted result table of 2016-17 Premier League Season using first 14-round games data

Comparing the top and the bottom of figure 15, we can see that the top 5 teams in the first 14 rounds stay consistent in the rest of the matches and Chelsea, who has outstanding home results and steady away performance win the champion of 2016-17. Tragically, the defending champion, Leicester city, performs bad in this season and could only strive to avoid relegation. Surprisingly, Spurs who ranks 5th in first 14 rounds seems to have huge potential in the next 24 rounds and finally win a 2nd rank while Arsenal seems to become laxity in the second half and wastes their accumulated advantages in the first 14- rounds.

VII. Future Developments

The project might be improved in those ways:

1. Our predictors only cover those naïve ones and more possible parameters that may have influence on the game results should be considered. For instance, the injuries situations is neglected in this project but is actually crucial in the real world. If haunted by a large amount of injuries, those top teams like Arsenal and Chelsea might face a hard time even when they are playing with those less competitive teams. Another nice example is the influence of schedule and other game events like champions league. If Manchester United has to play against Real Madrid in the weekdays on Champions league and has also to play against Arsenal at the weekend, the travel fatigue and the intense schedule might make them lose the game that they ought not to lose.

2. There might be other models that performs better than naïve Poisson regression model. For example, Hierarchical Bayesian Poisson Model introduced by Rasmus Baath seems to have full use of given data as prior information, and its detailed hierarchy might lead to more accurate results. Moreover, stepwise prediction that produces new estimation of parameters for the next step might also have a higher prediction accuracy.

3. One possible way to improve the accuracy in the current project is to augment the matrix and increase the data of those teams that are never relegated from the Premier League to figure out more accuracy estimators of those teams' parameters.

VIII. Conclusion

In this project, I discovered that the goals of each game follows Poisson distribution and figured out the least square estimator of Poisson parameter λs for each team by fitting the number of goals Y with a generalized linear model onto the team indicator matrix X. Based on this, I constructed two models with several modeling techniques and simulated the game results to obtain the predictions. Then, I designed three tests on two models and chose the best one for the prediction after an overall consideration of various aspects.

Within the best model, I successfully predicted the final result table of 2016-17 Premier League within the completed games. An overall observation indicates that my predicted results have a consistent trend as the completed games, which indicates that the result is reasonable and acceptable.

IX. References

1. Rasmus Bååth. “Modeling Match Results in Soccer using a Hierarchical Bayesian Poisson Model”. *http://www.sumsar.net/papers/baath_2015_modeling_match_resluts_in_soccer.pdf*
2. Pinnacle.com. *https://www.pinnacle.com/en/betting-articles/soccer/how-to-calculate-poisson-distribution*, Aug 12, 2014
3. Premier League website. *https://www.premierleague.com/*
4. Github user Jargnar. *https://raw.githubusercontent.com/jargnar/premier-league-data/master/2015-16/data.csv*