How Lucky Was Leicester City Winning the Premier League Season 2015/2016?

Jens Andreas Teigland Holck

@jensandreas.holck@gmail.com

Introduction

The 2015/2016 edition of the Premier League season was truly extraordinary as Leicester City was crowned champions, winning the Premier League for the first time in their history (see figure 1).

Position	Club	Played	Won	Drawn	Lost	GF	GA	GD	Points
1	Leicester City	38	23	12	3	68	36	32	81
2	Arsenal	38	20	11	7	65	36	29	71
3	Tottenham Hotspur	38	19	13	6	69	35	34	70
4	Manchester City	38	19	9	10	71	41	30	66
5	Manchester United	38	19	9	10	49	35	14	66
6	Southampton	38	18	9	11	59	41	18	63
7	West Ham United	38	16	14	8	65	51	14	62
8	Liverpool	38	16	12	10	63	50	13	60
9	Stoke City	38	14	9	15	41	55	-14	51
10	Chelsea	38	12	14	12	59	53	6	50
11	Everton	38	11	14	13	59	55	4	47
12	Swansea City	38	12	11	15	42	52	-10	47
13	Watford	38	12	9	17	40	50	-10	45
14	West Bromwich Albion	38	10	13	15	34	48	-14	43
15	Crystal Palace	38	11	9	18	39	51	-12	42
16	Bournemouth	38	11	9	18	45	67	-22	42
17	Sunderland	38	9	12	17	48	62	-14	39
18	Newcastle United	38	9	10	19	44	65	-21	37
19	Norwich City	38	9	7	22	39	67	-28	34
20	Aston Villa	38	3	8	27	27	76	-49	17

Figure 1: Final table of the 2015/2016 Premier League Season

In the 2014/2015 season, Leicester had finished 14th in the league and the season prior to that they had been promoted to the Premier League. This was truly a remarkable season that no one saw really saw coming. How could this be? And how much was luck involved? Sure, Leicester made some great signings and were fortunate in the sense that all of the top teams really disappointed this season. A couple of facts: Arsenal came closest with 71 points, the lowest ever for a runner-up. Leicester scored a total of 65 goals over 38 matches, averaging 1.71 goals. In comparison, Manchester City scored 102 goals when they won the 2013/2014 season. Leicester ended up with only 81 points, the fourth lowest score for a champion in all 24 seasons of the Premier League. So how lucky where they? I wanted to investigate how much the element of luck had been on Leicester's side throughout the campaign.

I decided to make a simple Poisson regression model and simulate the season 1000 times to see how many times Leicester would end up on top. To make things simple, I started with a data set containing all the results from the 2015/2016 season. The data consisted of four columns

home: the name of the home team
away: the name of the away team
yh: the score of the home team
ya: the score of the away team

The league consist of 20 teams that play each other two times, home and away for a total of $20 \cdot 19 = 380$ matches in a season. We will model this as follows: For each game, we assume that the number of goals scored between the home team and away team is independent. We assume that each team has a single parameter that measures its "strength". This is denoted $\beta_{Leicester}$ and so on for the other teams. If the home team is A and the away team is B, the score (number of goals) for team A will be Poisson distributed with mean λ, such that

$$\ln \lambda = \beta_0 + \beta_{home} + \beta_A - \beta_B,$$

where β_0 is the intercept and β_{home} is a home advantage parameter. Similarly, the score (number of goals) for team B will be

$$\ln \lambda = \beta_0 - \beta_A + \beta_B.$$

If two teams have equal strength, i.e. $\beta_A = \beta_B$, it follows that $\exp(\beta_0)$ will give the expected number of goals for the away team and $\exp(\beta_0 + \beta_{home})$ will give the expected number of goals for the home team.

Hence, the linear predictors are of the form

$$\begin{aligned} \ln E(Y) =& \beta_0 + \beta_{home} x_{home} + \beta_{Leicester} x_{Leicester} \\ &+ \cdots + \beta_{AstonVilla} x_{AstonVilla} \end{aligned}$$

Let A be the home team and B the away team. If the score Y is for A, then $x_{home} = 1$, $x_A = 1$, $x_B = -1$ and $x_C = 0$

for all other teams C. If the score Y is B, then $x_{home} = 0$, $x_A = -1$, $x_B = 1$ and $x_C = 0$ for all other teams C.Hence, there are $n + 1 + 1 = 22$ covariates. We construct a 720×22 design matrix X. However, the 20 team covariate vectors are linearly dependent. To work around this problem, we set one of the parameters equal to zero, $\beta_{AstonVilla}$. Aston Villas was chosen since it was the team that finished last. The covariate vectors will now be linearly independent. In order to calculate the parameters for the Poisson regression we must use maximum likelihood. For a Poisson regression, we have

$$L(\beta) = \prod_{i=1}^{n} \frac{\lambda_i^{y_i}}{y_i!} e^{-\lambda_i}.$$

The log-likelihood, $l(\beta)$ then becomes

$$l(\beta) = \sum_{i=1}^{n} y_i(\ln(\lambda_i) - \lambda_i - \ln(y!)).$$

Using our linear predictor and log link $\eta_i = ln(\lambda_i) = \mathbf{x}_i^T \beta$, the log-likelihood can be re-written as

$$l(\beta) = \sum_{i=1}^{n} y_i \mathbf{x}_i^T \beta - \sum_{i=1}^{n} e^{\mathbf{x}_i^T \beta} + C,$$

where $C = -ln(y!)$. Differentiating the log-likelihood function gives the score function,

$$s(\beta) = \sum_{i=1}^{n} (y_i - \lambda_i) \mathbf{x}_i$$

We use numerical optimization to solve

$$s(\hat{\beta}) = 0.$$

Results

Simulating the season 1000 times produced the following results as shown in figure 2. The second column shows the average number of points over 1000 seasons and the third columns shows the average goal difference.

```
##                   points goaldif
## Tottenham         73.562  33.576
## Leicester         72.873  31.879
## ManchesterCity    71.538  29.781
## Arsenal           70.731  28.691
## Southampton       64.082  18.027
## WestHam           62.089  14.786
## ManchesterUnited  61.524  14.247
## Liverpool         60.732  12.879
## Chelsea           56.167   5.897
## Everton           54.797   3.768
## Swansea           45.822 -10.071
## Watford           45.757  -9.683
## CrystalPalace     44.382 -11.932
## WestBromwich      43.389 -13.717
## Sunderland        43.180 -14.420
## Stoke             43.143 -13.932
```

Figure 2: Average league table of 1000 simulated seasons

Interestingly, this simple model shows that Tottenham has a higher average points score than Leicester City and ends up on the top of the table. Furthermore, Leicester gets roughly 73 points, which is 8 points lower than they actually got in 2015/2016, a considerable difference. Figure 3 shows the simulated results for each team with the vertical axis showing the number of seasons and the horizontal axis showing their place in the league.

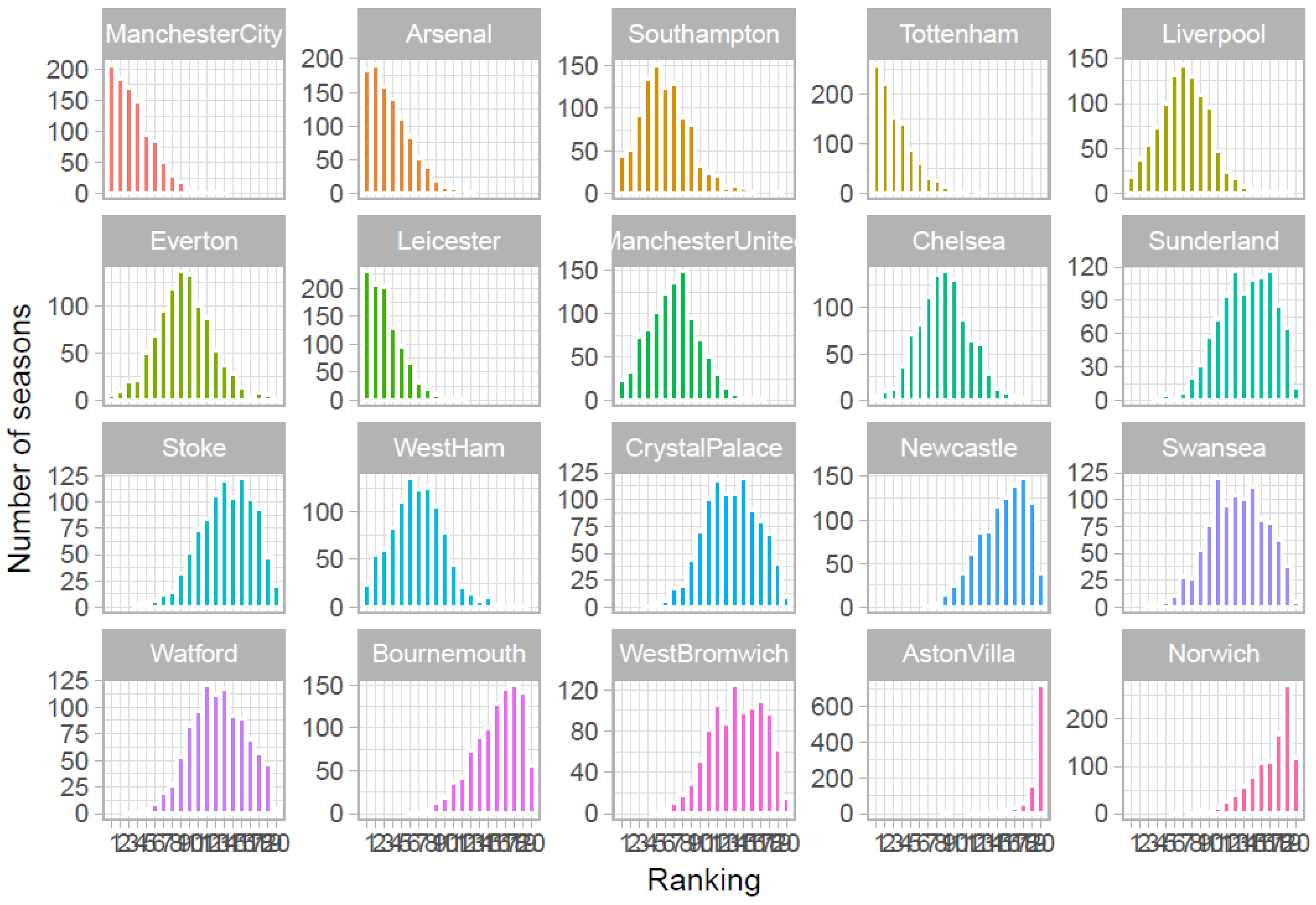

Figure 3: 1000 simulated seasons showing the final standings for each team.

Code

This simple model was made in R

```
generalizedLinearModel <- function(formula, data, contrasts = NULL, ...){
  # Extract model matrix & responses
  mf <- model.frame(formula = formula, data = data)
  X  <- model.matrix(attr(mf, "terms"), data = mf, contrasts.arg = contrasts)
  Y  <- model.response(mf)
  terms <- attr(mf, "terms")

  estimate = calc_estimate(X, Y)
  est <- list(terms = terms, model = mf, X=X, Y=Y)

  est$call <- match.call()
  est$formula <- formula
  est$estimate = estimate

  class(est) <- 'generalizedLinearModel'

  return(est)
}

log.lklh.poisson = function(par,x,y){
  return(sum(y * (x %*% par) - exp(x %*% par)))
}

calc_estimate = function(X, Y) {
  opt = optim(par=rep(0,17),
              log.lklh.poisson,
              x=X,
              y=Y,
              control=list(fnscale=-1),
              method=c("BFGS"))

  matrix = as.matrix(opt$par)
  rownames(matrix) = colnames(X)
  return(matrix)
}

final_table = function(input){
  result = data.frame(matrix(0L, nrow=length(levels(input$home)), ncol = 0))
  rownames(result) = levels(input$home)
  result$points = 0
  result$goaldif = 0
  for (i in (1:length(input$home))){

    home = as.numeric(input[i,]$home)
    away = as.numeric(input[i,]$away)

    goal_difference = input[i,]$yh - input[i,]$ya
    result[home, ]$goaldif = result[home, ]$goaldif + goal_difference
    result[away, ]$goaldif = result[away, ]$goaldif - goal_difference

    if(goal_difference > 0){
      result[home, ]$points = result[home, ]$points + 3
    }
    else if(goal_difference < 0){
      result[away, ]$points = result[away, ]$points + 3
    }
    else{
      result[home, ]$points = result[home, ]$points + 1
      result[away, ]$points = result[away, ]$points + 1
    }
  }

  result = result[(order(-result$points)),]
  return(result)
}
```