Homework 1 solution

netid: gp1655

1a

I created a "home" goal differential by subtracting away goals from home goals. I transformed the table into long format, and adjusted the differential based on whether a team was away or at home (if away: multiply goal differential by -1). Finally, I filtered for the EPL for the '17 season and used a pandas `groupby`, as well as three simple functions, to determine the average goal differential and the win/draw/loss count. Finally, I sorted the rows by their average goal differential `avg_goal_diff` in a descending manner. The following is the resulting table.

Team	game_count	avg_goal_diff	wins	draws	losses
Man City	38	2.07895	32	4	2
Liverpool	38	1.21053	21	12	5
Man United	38	1.05263	25	6	7
Tottenham	38	1	23	8	7
Chelsea	38	0.631579	21	7	10
Arsenal	38	0.605263	19	6	13
Burnley	38	-0.0789474	14	12	12
Leicester	38	-0.105263	12	11	15
Newcastle	38	-0.210526	12	8	18
Crystal Palace	38	-0.263158	11	11	16
Everton	38	-0.368421	13	10	15
Bournemouth	38	-0.421053	11	11	16
Southampton	38	-0.5	7	15	16
Brighton	38	-0.526316	9	13	16
Watford	38	-0.526316	11	8	19
West Ham	38	-0.526316	10	12	16
West Brom	38	-0.657895	6	13	19
Swansea	38	-0.736842	8	9	21
Huddersfield	38	-0.789474	9	10	19
Stoke	38	-0.868421	7	12	19

1b

I took the table from exercise 1a and created the points column by multiplying wins by 3 and adding to the number of draws for each team. I sorted the table by descending point totals. The resulting table is:

Team	game_count	avg_goal_diff	wins	draws	losses	total_points
Man City	38	2.07895	32	4	2	100
Man United	38	1.05263	25	6	7	81
Tottenham	38	1	23	8	7	77
Liverpool	38	1.21053	21	12	5	75
Chelsea	38	0.631579	21	7	10	70
Arsenal	38	0.605263	19	6	13	63
Burnley	38	-0.0789474	14	12	12	54
Everton	38	-0.368421	13	10	15	49
Leicester	38	-0.105263	12	11	15	47
Crystal Palace	38	-0.263158	11	11	16	44
Bournemouth	38	-0.421053	11	11	16	44
Newcastle	38	-0.210526	12	8	18	44
West Ham	38	-0.526316	10	12	16	42
Watford	38	-0.526316	11	8	19	41
Brighton	38	-0.526316	9	13	16	40
Huddersfield	38	-0.789474	9	10	19	37
Southampton	38	-0.5	7	15	16	36
Stoke	38	-0.868421	7	12	19	33
Swansea	38	-0.736842	8	9	21	33
West Brom	38	-0.657895	6	13	19	31

1c

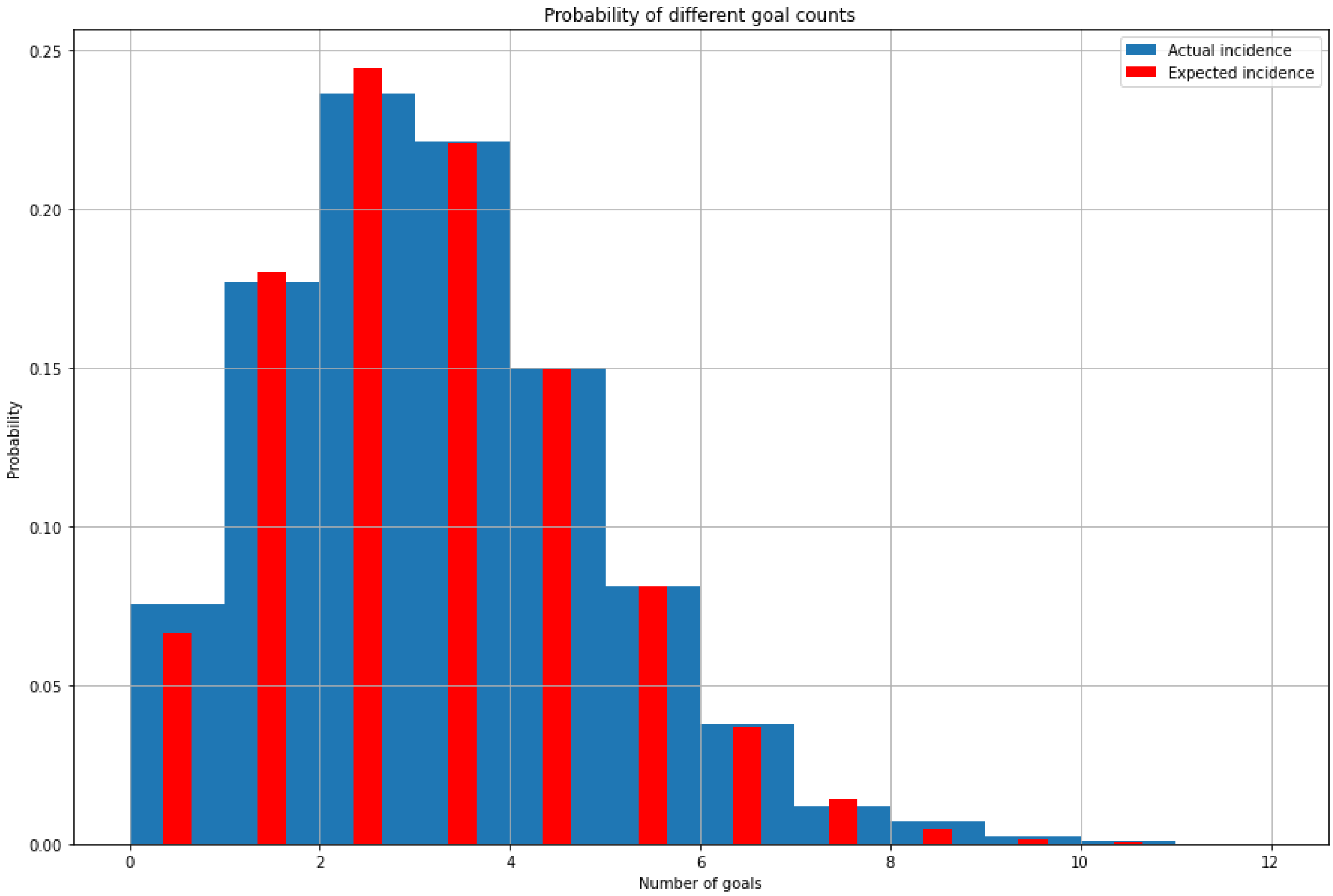
For this exercise, I took the long intermediate table from exercise 1a and filtered for the '17 season, using a similar `groupby` as for 1a. I sorted the table by descending average goal differential (instrumental for the next transformation). I then created a `GroupBy` object, grouping by division, and used the `.head()` accessor to take the first 3 values (which are the highest average goal differentials per division). This process only yielded the highest average goal differentials per division because of the previous sorting. Finally, I sorted the table as instructed. The result follows:

Div	Team	game_count	avg_goal_diff	wins	draws	losses	total_points
Bundesliga	Bayern Munich	34	1.88235	27	3	4	84
Bundesliga	Hoffenheim	34	0.529412	15	10	9	55
Bundesliga	Dortmund	34	0.5	15	10	9	55
EPL	Man City	38	2.07895	32	4	2	100
EPL	Liverpool	38	1.21053	21	12	5	75
EPL	Man United	38	1.05263	25	6	7	81
La_Liga	Barcelona	38	1.84211	28	9	1	93
La_Liga	Real Madrid	38	1.31579	22	10	6	76
La_Liga	Ath Madrid	38	0.947368	23	10	5	79
Ligue_1	Paris SG	38	2.07895	29	6	3	93
Ligue_1	Lyon	38	1.15789	23	9	6	78
Ligue_1	Monaco	38	1.05263	24	8	6	80
Serie_A	Juventus	38	1.63158	30	5	3	95
Serie_A	Napoli	38	1.26316	28	7	3	91
Serie_A	Lazio	38	1.05263	21	9	8	72

2a

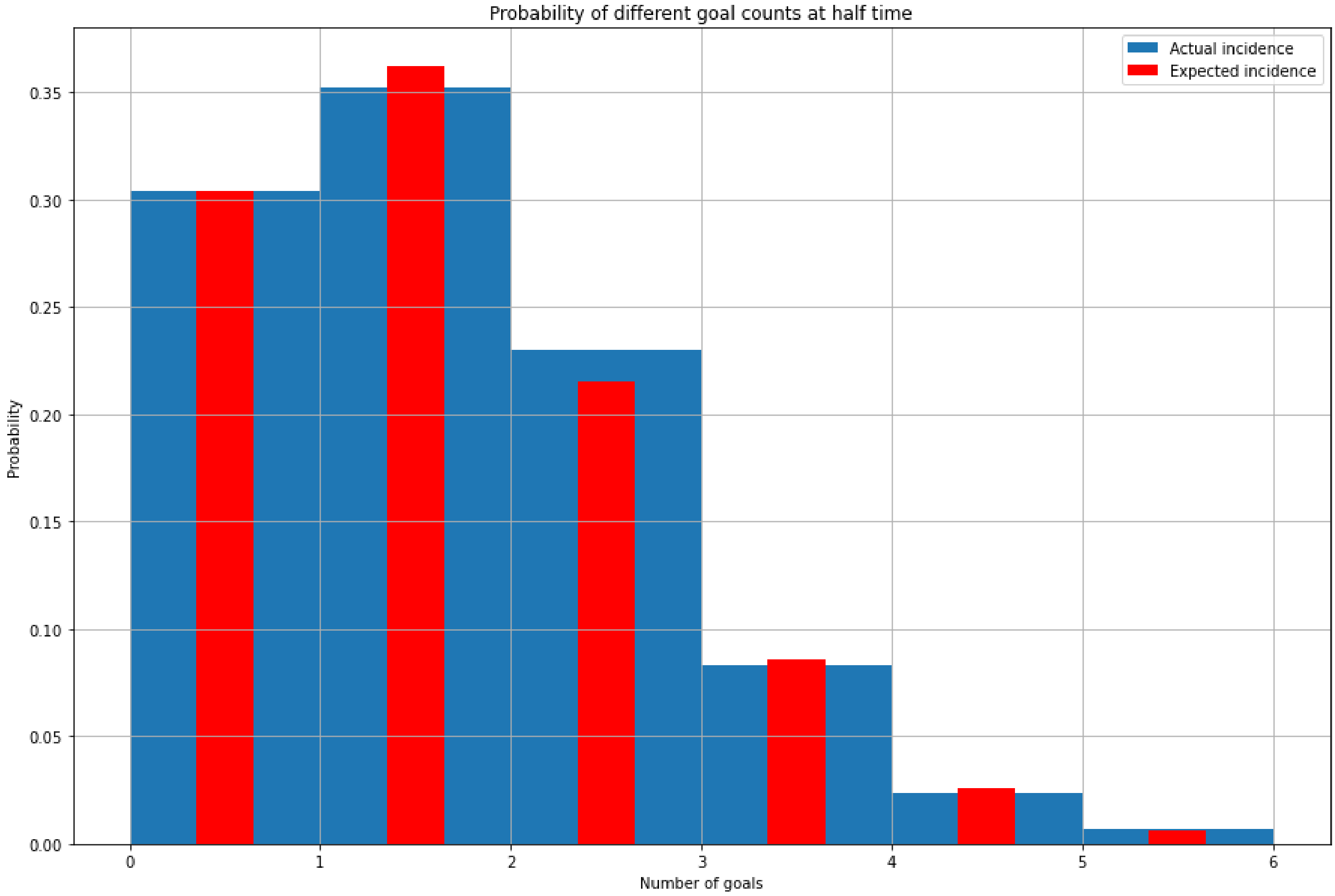
I used a Poisson distribution for modeling. This is due to two reasons: first, I already knew from prior experience modeling points in sports like this works very well with a Poisson distribution (from, for instance, Maher 1982, as well as discussions with a friend who works in sports analytics, and having used Poisson distributions myself to model hockey scoring). Second, I looked at a histogram of the total goals in a game and the histogram looked very similar to a Poisson distribution.

I used MLE to determine the λ parameter of the distribution. This means I simply took the average of the total goals per game and set that as λ. For 2a, this parameter was $\lambda_{2a} \approx 2.71$. Plotting the histogram and the PMF of the chosen Poisson distribution yields the following graph.



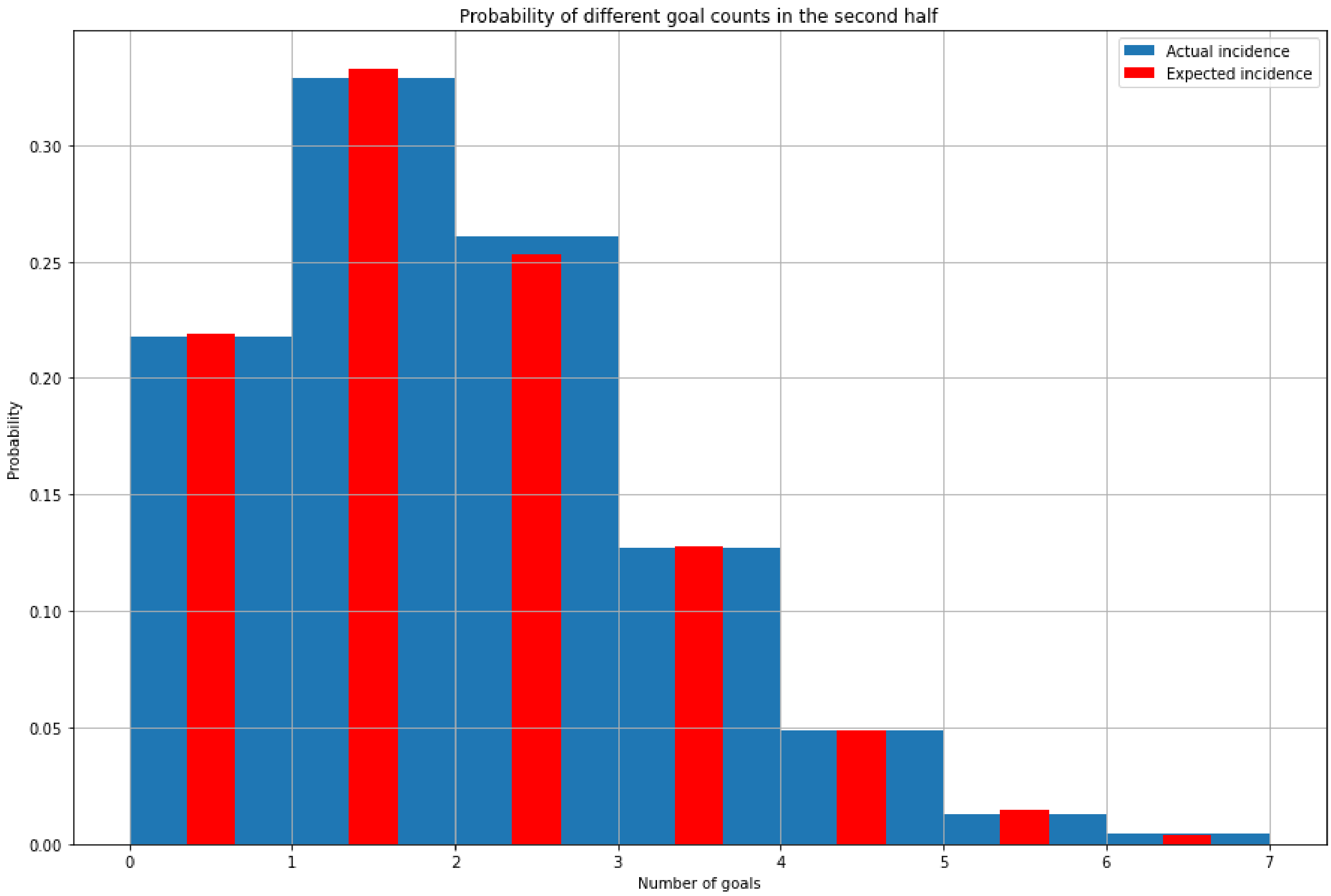
2b

I used an identical process to 2a here. The Poisson distribution parameter was $\lambda_{2b} \approx 1.19$. The following graph resulted:



2c

I once again used the same process as 2a and 2b. The Poisson distribution parameter was $\lambda_{2c} \approx 1.52$. The following graph shows the result.



2d

For this exercise, I used the wide dataset format. I simply grouped by division, counting over the number of rows (since each row is a game) and taking the average of the total goals scored in a game. I sorted the table by descending average goals per game. The following table shows the result.

division	number_of_games	avg_goals
Bundesliga	1224	2.81127
La_Liga	1520	2.75921
Serie_A	1520	2.72566
EPL	1520	2.68618
Ligue_1	1520	2.58816

2e

According to the approach described in class, we expect similar-odds teams with a given number of total goals to follow a binomial distribution. For 4 goals, the possible outcomes are: 4-0, 3-1, 2-2, 1-3, 0-4. Considering the odds of a goal being a coin flip (i.e. 50%), then the odds of a 2-2 outcome are

$$\binom{4}{2} 0.5^2 \cdot 0.5^2 = \frac{6}{16}$$

I selected the games from the dataset where: firstly, the total goals scored in the game was 4; and secondly, the market-implied probability of the home team winning and the implied probability of the away team winning deviated by less than 0.04, since 0.02 yielded only 37 games and I thought that was too few. Using 0.04, I got 63 games. As a result, we then expect

$$63 \cdot \frac{6}{16} = 23.625$$

games to end in a draw. Rounded up, this is 24. To determine the actual number of games that ended in a draw, I simply counted the number of games in this data subset that had an equal number of full-time home goals and full-time away goals (which, with 4 goals in total, corresponds to games that ended in a 2-2 draw). This yielded an actual count of 33. Calculating the standard deviation from this is simple:

$$\sigma = \sqrt{np(1-p)} = \sqrt{63 \cdot \frac{6}{16} \cdot \frac{10}{16}} = 3.843$$

(where the Bernoulli p is $\frac{6}{16}$ because we are evaluating the Bernoulli "is draw"/"is not draw")

$$\frac{33 - 24}{3.843} = 2.34$$

Since we are more than 2 standard deviations away from the "expected" number of draws, we can conclude that we observe a very strong comeback tendency in this dataset.