# PHSX 815 Project 1:
# Quantifying a Soccer Team's Performance:
# Simulation and Hypothesis Testing of the Average Number of Goals Scored by a Team Per Game

Ashley Lieber

February 13, 2023

## 1 Introduction

As the year 2022 drew to a close, much of the world had their eyes on the television screen to see which country's soccer team would rise to the top and win the FIFA World Cup. This event captivated millions across the globe and serves as the inspiration for the simple simulation described in this paper.

This paper follows a simple simulated experiment in which one would observe many games played by a single soccer team over many seasons and record the number of goals scored by the team for each game. From this data, one could calculate the average number of goals scored by the team. The higher the average, the more goals that the team usually scores. The more data that is gathered, the more accurate the average will be. Instead of spending the time and effort to record this data, the code in this paper can simulate the distribution of data that we could expect given a certain rate. One could simulate this data for any average their heart desires (e.g. 2, 5, 10, etc.), but the question remains, what average best describes a team's performance? This is where hypothesis testing will come to play. The code outlined in this paper will simulate a data set generated using two different averages and then perform hypothesis testing to understand how well one could distinguish which average is more reflective of the observed team's performance.

This paper is organized as follows: Sec. 2 explains the two hypotheses we are testing. Sec. 3 describes the Python computer simulation that was developed to simulate the data for these two hypotheses. Next, Sec. 4 explains how these outputs were analyzed and interpreted. Lastly, the conclusions of this simulation are presented in Sec.5.

## 2 Hypotheses Explanation

For the purposes of this paper, we will outline the testing of two potential hypotheses given this scenario. The only difference between these two hypotheses is the assumed rate of goals scored per game that is used to generate the data. This rate is called the "configurable parameter" of the experiment and will be denoted by the Greek letter $\lambda$. The first hypothesis that will be tested is denoted by $H_0$ and assumes that the team scores an average of three goals per game ($\lambda = 3$). The second hypothesis that will be tested against the first hypothesis will be denoted as $H_1$ and assumes that the team scores an average of five goals per game ($\lambda = 5$). With these hypotheses in hand, the code will generate a data set for each

hypothesis and subsequently analyze this data which will aid in determining which hypothesis is more likely to reflect the true average for the soccer team in question.

## 3 Code and Experimental Simulation

First and foremost, to conduct this analysis, a data set for each hypothesis is needed. To obtain these, the code will randomly sample data from a Poisson distribution. This means the code will simulate the act of watching a number of games and recording the number of goals scored by the team. Based on this scenario, a single measurement can take on values from $[0, \infty)$, but not negative values since a team cannot score negative goals. The data will also be sampled randomly from a Poisson distribution. This distribution was carefully chosen because the characteristics of our scenario meet the criteria for a Poisson distribution. These criteria are as follows: (1) the "individual events occur at random and independently in a given interval (this interval can be of time or space," and (2) "the mean number of occurrences of events in an interval (time or space) is finite and known" represented by the symbol $\lambda$ [1]. This particular experiment fits these criteria because the measurements of goals scored by a team are random events recorded once per game and each game is in and of itself an independent event. Additionally, the average number of occurrences, $\lambda$ is a finite number and known under each hypothesis. Furthermore, the Poisson distribution is a "discrete probability distribution that describes probabilities for counts of events [2]. Thus, the Poisson distribution is an apt choice to randomly sample our data sets from The Poisson Probability equation is given by the following formula which calculates the probability of x occurring in a given interval

$$P(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

where $\lambda$ is the average number of measurements per interval, $e$ is the constant Euler's number which is approximately $2.78$, and $x$ which takes discrete values and corresponds to our data [3].

With the correct distribution chosen, parameters defined, and hypotheses set, the code can now be used to randomly sample data according to the Poisson distribution using the code $GoalData.py$. This code takes in the rate parameter that we have defined for the hypothesis, as well as parameters that define the number of measurements to be taken per experiment. The best way to think of this organization within our given scenario is how many games we expect to observe per season, and how many seasons we want to observe. Since we are simulating the experiment rather than actually taking data ourselves, we can choose to simulate as many measurements and experiments as we would like. The more data that is collected the more each distribution will center around and reflect the $\lambda$ the data set was generated on. In this case we chose the number of measurements (or games ) observed to be $N_{meas} = 20$ to reflect the average number of games in a college soccer season and the number of experiments (or seasons) observed to be $N_{exp} = 10,000$. With this information, the first script $GoalData.py$ can now be run which will sample all of these data points randomly from a Poisson Distribution using the equation shown earlier and the external SciPy package [4]. For each measurement needed, the code will sample a number from the Poisson distribution. This utilizes a nested for loop to sample each measurement. The resulting data will be a discrete value that is a non-negative integer $(e.g. 0, 1, 2, ...)$ [2]. This code will be run twice first to generate the data set that corresponds to the $H_0$ hypothesis that $\lambda = 3$ and again to generate the data set that corresponds to the $H_1$ hypothesis that assumes $\lambda = 5$. These data set results are then saved as a text file which is a persistent data format. The format of this document is that the first line lists the $\lambda$ used to generate the data, and then each subsequent line holds an array of all the measurements for a single experiment. For example, if we sampled data with $\lambda = 2$ with $N_{meas} = 5$ and $N_{exp} = 10$ the data would appear in the format shown in Figure 1.

```
(base) ashleylieber@PHSX-MILLS-22 PHSX815_Project1 % python3 GoalData.py -rate 3 -Nmeas 5 -Nexp 10
3.0
2 4 7 5 3
5 4 2 3 1
5 1 0 3 3
0 1 1 2 3
0 1 2 4 6
1 4 4 1 3
5 3 2 5 3
3 1 5 0 1
3 4 4 6 3
1 3 6 4 5
```

Figure 1: Example of simulated data output if $\lambda = 2$, $N_{meas} = 5$, and $N_{exp} = 10$

The data that is simulated for each hypothesis is then sent to another Python script which is titled $GoalDataAnalysis.py$ which further analyzes the data and performs the hypothesis tests needed.

## 4 Analysis

After generating the randomly sampled data for each hypothesis, the analysis can begin. First, the data files are read in to ascertain the rate that was used for the code as well as to put each individual measurement into a list. This occurs for each data set. Next, the $25\%, 50\%$, and $75\%$ are calculated for each data set. Note: the $50\%$ is the same numerically as the median value [5]. Then to visualize the data each data set is plotted as a histogram with lines drawn to show each quantile calculated. Please see Figure 2 for the resulting histograms for the data set considered here with $\lambda = 3$ and $\lambda = 5$.

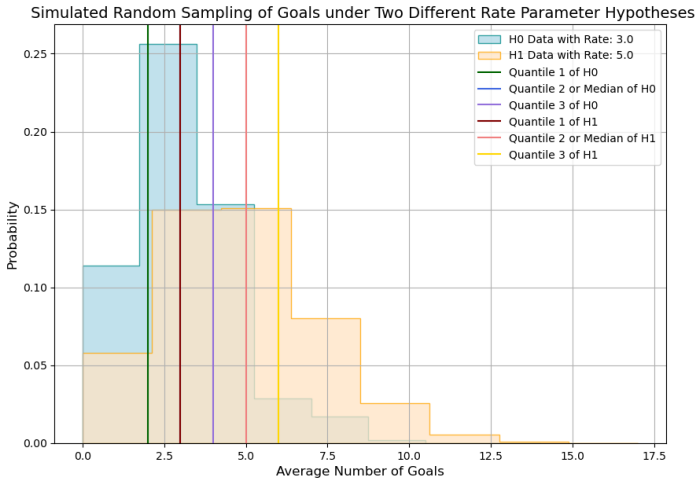


Figure 2: Simulations of the average number of goals per game over 10000 observations under two different hypotheses. The data shown in blue is the average number of goals under the hypothesis that the rate parameter is $\lambda = 3$ goals per game. Analogously, the data distribution shown in orange corresponds to the average number of goals under the hypothesis that the rate parameter is $\lambda = 5$ goals per game. This data was sampled from 20 measurements per 10,000 experiments totaling 200,000 total observations for each hypothesis. In terms of the scenario, this would correspond to observing 20 games (a season) over 10,000 times. Shown in the figure are the 0.25, 0.50 (median), and 0.75 quantiles. Note: the Median of the H0 data and the 0.25 Quantile of H1 are the same, thus only one line is visible.

The next step in the analysis will allow us to conduct an actual hypothesis test to quantitatively define how well separated our data distributions are which directly corresponds to our ability to differentiate between the likelihood of either hypothesis. The first step in this process is the calculate the log likelihood ratio (LLR) for each experiment in each hypothesis and store it as an array. The equation below will be used to calculate this log likelihood ratio.

$$\Lambda = log(L_x) = log\left(\frac{P(x_i|\lambda_1)}{P(x_i|\lambda_0)}\right) = log\left(\frac{\prod_i^{Nmeas}(Pois(x_i|\lambda_1)}{\prod_i^{Nmeas}(Pois(x_i|\lambda_0)}\right)$$

The uppercase lambda $\Lambda$ is known as the test statistic. The $Pois(x_i|\lambda_1)$ refers to the Poisson probability of the data point given the rate in hypothesis $H_1$, and the converse is true when $\lambda_0$ is used. In order to calculate the Poisson probability used in the equation, the method $poisson.pmf$ from $Scipy.stats$ [4] was used which calculated the Poisson probability given the data point and the rate value.

At this point, the code has arrays of the log likelihood values for each hypothesis in an array. I utilized the method of Bubble Sorting [6] to sort these arrays. Please note that this step can take quite a long time when conducting many measurements and experiments so be mindful of that when running this program. Each of these arrays are now sorted into ascending order so that further statistics and figure plotting can be conducted with them. The plot of these log likelihood ratios or test statistics is shown in Figure 4.

After sorting has concluded, the hypothesis testing can finally begin. The first step in the process for this next stage is to define a significance level $\alpha$ for our test. In this case, the values of $\alpha = 0.02$ was chosen which corresponds to the test being accurate $98\%$ of the time (a confidence level of 0.98 or $(1-\alpha)$). This sets our false positive rate also known as a Type 1 error. This means that we will tolerate a false positive rate $2\%$ of the time. The $\alpha$ parameter is also chosen ahead of time in an effort to avoid experimental bias. A visual representation of this parameter is shown in Figure 3

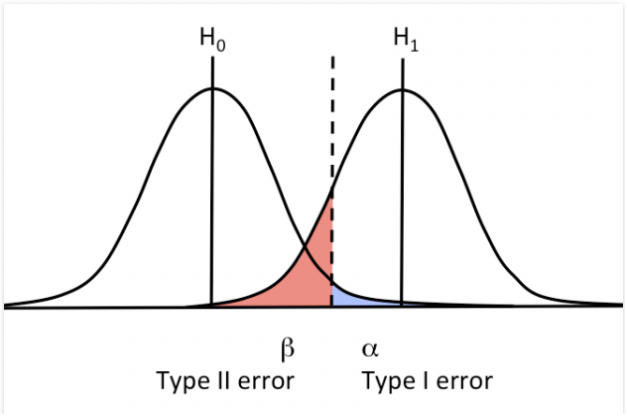


Figure 3: Figure depicting how $\alpha$ and $\beta$ are determined from a graph. The dashed line would correspond to the critical $\Lambda$ in this case. [7]

Using this value for $\alpha$, we will find a test statistic value in the array for $H_0$ which corresponds to $98\%$ of the area under the curve to be one one side and $2\%$ of the area being on the left. Essentially, we find the quantile for 0.98. This value is known as the critical test statistic and this value for the data we have simulated is sown in Figure 4 as a red line. The next step is to define the value for $\beta$ which as can be seen from Figure 3 is the area for the other distribution (in this case $H_1$) that is sectioned off by

the critical $\Lambda$ value. This is known as a Type II error which corresponds to our false negative rate. For this case, the value for beta ($\beta$) was determined to be 0.005 which means that the line did not section off much data for the $H_0$ distribution. The value for $\beta$ helps us to quantify the Power of the Test which is calculated simply with the expression $1 - \beta$. In this case, the power of the test is 0.995. This value of $\beta$ shows a very powerful test since the false negative rate is very low. The most powerful test would correspond to a value of 1.

Finally, all of these analytical concepts are combined and displayed in Figure 4. This concludes the formal

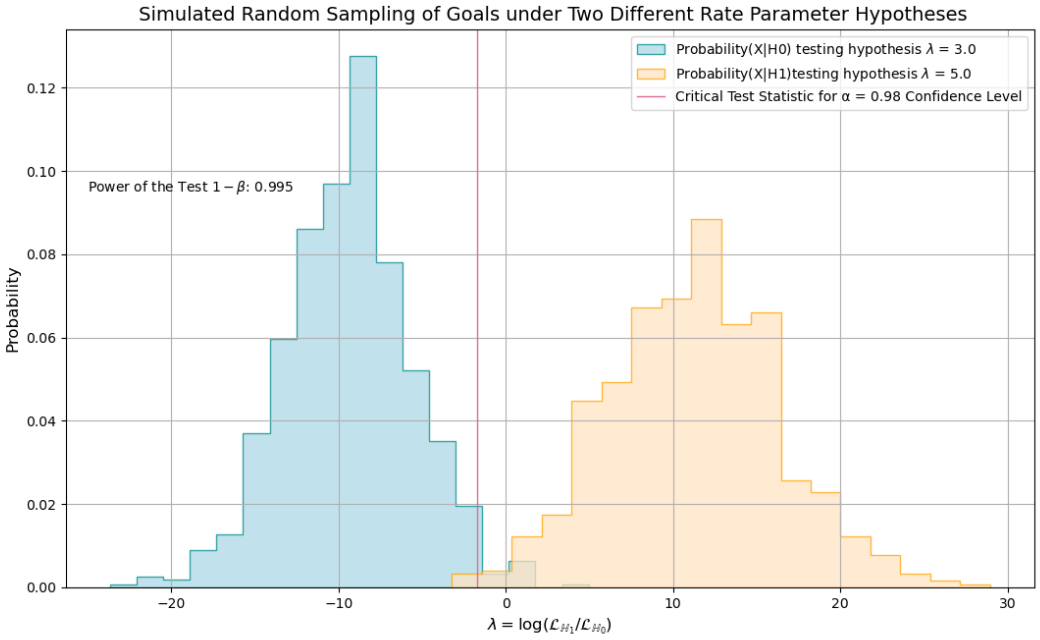


Figure 4: Final log likelihood ratio or test statistic plot showing the distribution of our two hypotheses. The critical $\Lambda$ is also displayed in red along with the power of the test displayed in the top left corner of the plot. The blue histogram corresponds to the distribution from the data for hypothesis $H_0$ and the orange distribution corresponds to hypothesis $H_1$

hypothesis testing that the code conducts and now the results can be interpreted under the scenario that we crafted. Based on the output in Figure 4 we see that the distributions have very little overlap and based on our value for $\beta$ we have conducted a very powerful test, most likely too powerful for our purposes. This means that we can distinguish between the two hypotheses very well. Since $\lambda$ can only take on values that are whole numbers (e.g. there's no such thing as a "half-goal"), the fact that we have such a powerful test when the difference between the two rate parameters $\lambda_1 - \lambda_0$ is only two, means that there is only one comparison that could be closer and thus theoretically harder to distinguish. Namely, this additional test would compare rate parameters that are only one count apart (e.g. $\lambda = 3$ and $\lambda = 4$). If we wanted to improve the power of the test we could either choose less likely hypotheses to test (e.g. $\lambda = 2$ vs. $\lambda = 25$), or we could take more data. Since, this hypothesis test ended with such a powerful test, it begs the question, did we realistically need to take so many trials to distinguish these two hypotheses? Most likely not. However, to obtain the power of the test at the significance we stated, that is how many trials it took.

Up until this point, we have only considered our two chosen hypotheses, but not how this test can be used to make determinations about which rate value best corresponds to the team's performance. In order to make use of this hypothesis test, we would need to take a sample of real-world data from our team. For example, we could observe an entire season of games, recording the number of goals for the team for each game. From this data, we could compute the log likelihood ratio or test statistic as seen above. We could then plot this test statistic on Figure 4. If this test statistic is less than the critical test statistic, then we can say that the true rate for the team is more likely to be $\lambda = 3$ than it is to be $\lambda = 5$. The converse is also true. If the test statistic calculated for the real world data is greater than the critical test statistic, then the true rate for the soccer team is more likely to be $\lambda = 5$ than it is to be $\lambda = 3$. This is how a simulation and hypothesis test such as this one can be used to make statements about real world data and scenarios.

# 5 Conclusion

Overall, this simulated experiment demonstrates how data for two competing hypotheses can be randomly sampled according to a Poisson distribution, analyzed, and interpreted according to our soccer team scenario. We were able to compare data from two opposing hypotheses to determine which might be more likely depending on a real world data set. Even without the real world data set present, we are able to quantify the separation of the distributions through the value of $\beta$ and the power of the test under a certain significance level $\alpha$. This code demonstrates how we can statistically determine, to a certain level of confidence, what the true rate of goals per game is for a particular soccer team. So next time you tune in to watch your favorite soccer team compete or are lucky enough to watch in person, think about taking goal data to apply this method in quantifying their performance for the season.

# References


[1] A. Kumar, *Poisson distribution explained with python examples*, Oct, 2021.

[2] J. Frost, *Poisson distribution: Definition amp; Uses*, May, 2022.

[3] O. Eaton, *Modelling the Distribution of Football Goals*,.

[4] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, *Nature Methods* **17** (2020) 261–272.

[5] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, *Array programming with NumPy*, *Nature* **585** no. 7825, (Sept., 2020) 357–362.

[6] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, , *Numerical recipes: The art of scientific computing*. Cambridge University Press, 2020.

[7] D. Darrin and D. Darrin, *Type I and type II error*, Oct, 2022.