PHSX 815 Project 4: Poisson Statistics and Photon Counting: An Application of Poisson Data Simulation and Statistics to Real World Physics Experiments

Ashley Lieber

May 8, 2023

1 Introduction

Over the course of this semester, my projects have primarily focused on Poisson statistical simulations and experiments of increasing complexity. Each project has focused on simulating many soccer games to evaluate different models of a soccer team's performance (i.e. number of goals scored per game over many games and seasons). While this is an interesting query for those interested in soccer statistics, there are many other scenarios these models can be applied to especially within the realm of physics. In essence, any experiment that deals with counting, a Poisson distribution is an apt choice to describe the distribution of the data. The purpose of this project is to explore a simple photon counting experiment. The simulation will model the number of photons from a coherent laser that hit a detector in a particular time interval which will be described by a Poisson distribution [1]. The rate at which these photons hit the detector is based on the power of the laser, however, the efficiency of the detector should also be factored in. This efficiency is unknown to the scientist conducting the experiment thus the analysis in this project would be useful to them in understanding the limitations of their experimental setup.

This paper is organized as follows: Sec. 2 explains the setup of the photon counting experiment, explores the Poisson model and data generation for the simulation, Sec. 3 walks through the analysis code and the computations it performs on a given data set. Next, Sec. 4 explains the different scenarios that were tested in this experiments and the findings that can be taken from those tests. The following Sec. 5 presents an overall summary of the conclusions of this simulation. Lastly, Sec. 6 provides the link to the GitHub repository for this project which contains all pertinent code scripts, referenced data sets, and figures as well as instructions on how to use and understand the various items.

2 Photon Experiment Setup and Data Simulation

2.1 Laser Photon Counting Experiment

In this project, we are considering a scientist who has set up a photon counting experiment. This type of experiment can be applied in many scientific fields including physics, materials science, and chemistry. In this paper, we will be considering the setup where a scientist – who we will call Ken – has an optical set up where a laser is pointed at a sensitive photo detector. He hopes to perform experiments with this device, but first he would like to collect all the information about the detector such as the rate at which

it collects data, the power of the laser he is using, and the efficiency of the detector since he knows those will be critical values in his later calculations. He was able to learn the lasers power and the rate at which the detector records observations, but he cannot find out what the detector's efficiency is. Furthermore, Ken has lost the instruction manual and the manufacturer, who he has emailed to ask, has been radio silent. Ken, did not let this get him down, rather, he has devised an experiment in which he will be able to record the average number of photons per second, and from that value, he can devise the efficiency of his detector.

2.2 Poisson Model Selection and Setup

In photon experiments, they are often called counting experiments because of the particle nature of light. A laser is a stream of photons hitting the detector and each arrival of a photon can be considered a count by the photo-detector. This means that we can characterize a laser by the number of photons per second. Each of these counts can be considered a random and individual event. Thus when considering what model might best describe data of this type, the Poisson distribution is a natural choice [1]. In fact, a coherent light source, such as a laser, can be exactly modeled by a Poisson distribution. Other light sources such as LED (which are less coherent) adhere to Super-Poisson distributions, and others such as quantum light, could in theory, be explained with Sub-Poisson statistics [2]. To further confirm that the Poisson is an apt choice we will explore the clear stipulations it requires be met. In order to use a Poisson distribution, several criteria must be met which are as follows: (1) the "individual events must occur at random and independently in a given interval of time or space", and (2) "the mean number of occurrences of events in an interval (time or space) is finite and known, [3]. As we have discussed, the individual photon counts by the detector can be considered both random and independent events that are separated by an interval of time. Additionally, the "mean number of occurrences" or the average rate of photons hitting the detector is a finite number that can be known. This average rate is often denoted by the lowercase Greek letter λ. Additionally, a single measurement can only take values from $[0, \infty)$ which also fits this model selection and the scenario since we cannot detect negative a negative number of photons. Beyond simply meeting the criteria of Poisson distributions, it is known in statistical practices that the Poisson distribution is an apt choice for event data as it is a "discrete probability distribution that describes probabilities for counts of events" [4]. The Poisson Probability Distribution equation is given by the following formula which calculates the probability of data point, x, occurring in a given interval based on some rate parameter λ

$$P(x_i|\lambda) = \frac{\lambda^x e^{-\lambda}}{x!} \tag{1}$$

where λ is the average number of measurements per interval (rate), e is the constant Euler's number which is approximately 2.78, and x which takes discrete values and corresponds to our data [5]. An example of what a Poisson distribution looks like is shown in Figure 1 which corresponds to a rate of 3 which means three photons reach the detector every second. .

So, now that an overarching model has been , in simulating data for this experimental set up, we will randomly sample data from a Poisson distribution based on a given rate of photons per second. The next subsection 2.3 explore this in greater detail. .

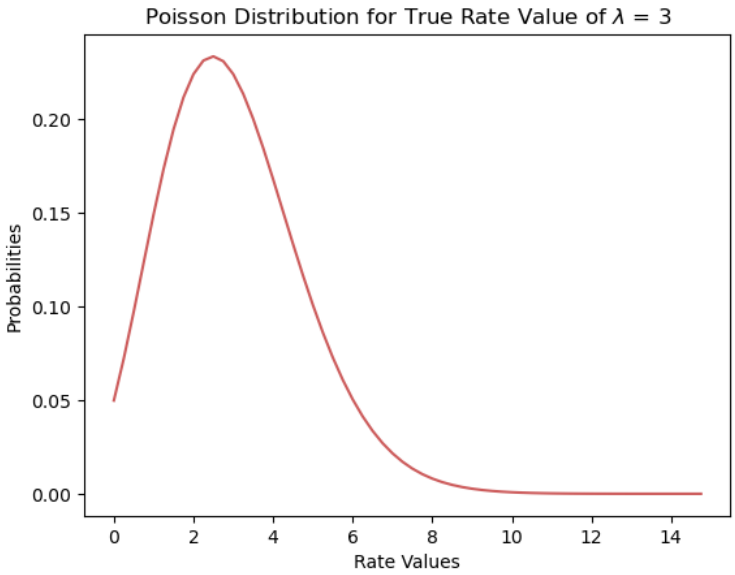

Figure 1: Example Poisson Distribution with a rate value of $\lambda = 3$. This is an example of the distribution from which data is randomly sampled from.

2.3 Data Generation

The data can be generated according to this Poisson distribution by randomly sampling data using the code script named $ExperimentData.py$ which can be found in the repository (linked in 6). When running this script, there is the ability to specify a number of parameters in order to generate data for different scenarios. These parameters are the number of measurements (Nmeas), the number of experiments (Nexp), seed, power of the laser in Watts (laserpower), the efficiency of the detector (detect-eff), the mesurement time in seconds (Tmeas) as well as the filename to save the dataset to.

In the context of the our photon experiment, this means that Ken could simulate the data from his experimental set up by inputting the power of his laser, the measurement time of the detector as well as the number of measurements and experiments he would like to simulate. However, the the case where Ken would be doing this in real life, he would simply use his set up to take the data and then use the same analysis measures. We are simply providing a way to simulate that data collection in a somewhat realistic manner. 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. An advantage of this is simulating a large number of experiments that one might not realistically be able to do! For example, the dataset $ExpData - 0.9DE - 1000M - 10000E.txt$, can be found in the Project 4 repository 6 which was a simulated dataset in this project. This data set is based on a laser power of 0.1 Watts, a detector efficiency of 0.90, a measurement time of 0.001 attoseconds, and finally 1000 measurements for each 10,000 experiments that were simulated. I should note that the extremely small measurement time was chosen to reduce the magnitudes of the numbers in the dataset. On this time scale, the simulation returns photons per second on the scale of hundreds. This is a byproduct of the fact that light moves incredibly fast at $3,000,000$ meters per second, so the number of photons can become incredibly large very quickly. The values that the user inputs are then used to calculate the rate of photons per second for the experiment's laser using the following equation [2].

$$E = nhc \tag{2}$$

$$n = \frac{E}{hc} = \frac{(laserpower) * (detectoreff iciency) * (Measurementtime)}{h * c} \tag{3}$$

where h is Planck's constant, E is energy, c is the speed of light, and n is the number of photons per second, our calculated rate parameter.

In order to sample values from a Poisson Distribution, the script $ExperimentData.py$ utilizes the external package Scipy.Poisson [6] which encodes the equation shown earlier which helps to condense and simplify our code since there was no need to write the algorithm again from scratch. 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 $(e.g. 0, 1, 2, ...)$ [4]. These measurement values are then stored in a persistent data file format (.txt) which can then be read in by the analysis program.

That data file, along with others found in the repository, demonstrates how the output of the data generation code is saved to be used in further analysis.

3 Analysis Methods

After generating the randomly sampled data, the analysis can begin which can be found in the file $MeasAnalysis.py$ in the repository. The analysis code can handle a single data file at a time, so this section will outline the algorithms and computations that are performed in order to estimate the true rate parameter for the given data set. First the code gathers a few parameters from the data set such as the true rate parameter, number of measurements, and number of experiments. The remainder of the analysis in order to estimate the rate parameter follows the method of maximum likelihood estimation (MLE). This maximum likelihood estimate of λ is the value of λ that maximizes the likelihood – "that is, makes the observed data 'most probable' or 'most likely' [7]. The first step of this method is write down a function of the parameter of interest (λ) that is proportional to the likelihood of the function given some data. The following equation shows how this likelihood was written down and simplified.

$$L(x) = \prod_{i}^{Nmeas} (Pois(x_i|\lambda) \tag{4}$$

$$= \prod_{i}^{Nmeas} \frac{\lambda^x e^{-\lambda}}{x!} \tag{5}$$

We can then take the log of this likelihood function in order to get it in a format that is more amenable to functioning within the code. The following derivation shows how this was achieved.

$$Log(L(x)) = Log(\prod_{i}^{Nmeas} \frac{\lambda^x e^{-\lambda}}{x!}) \tag{6}$$

$$= \sum_{i=1}^{Nmeas} (X_i log\lambda - \lambda - logX_i!) \tag{7}$$

$$= log(\lambda) \sum_{i=1}^{Nmeas} X_i - (Nmeas)\lambda - \sum_{i=1}^{Nmeas} log(X_i! \tag{8}$$

The value that will be the most likely estimate of the λ parameter is the value that maximizes this function, or equivalently, minimizes the negative of the log likelihood function. Since I preferred to use a minimization routine within the code, I opted to minimize the negative of the log likelihood function [8].

$$\lambda_{estimate} = argmin\left[- (log(\lambda) \sum_{i=1}^{Nmeas} X_i - (Nmeas)\lambda - \sum_{i=1}^{Nmeas} log(X_i!)\right] \tag{9}$$

$$\tag{10}$$

Within the analysis code, a value for λ is estimated for each experiment and stored as a list. The calculated values of the negative log likelihood estimate is also stored in an array to keep track of those values. Lastly, each data point is stored in a list so that we can visualize the spread of the data in tandem with the rate parameter estimations.

In order to estimate the rate parameter λ for a whole data set, rather than each experiment, the analysis utilizes and compares two different estimation methods that make use of the data stored in various lists as described in the earlier paragraph. These methods also allow us to estimate the uncertainty or error within our analysis to quantify the confidence in the final result. The first method is to create a histogram of the λ estimations and analyze the spread of the data. This distribution should peak at the value of the true rate parameter (λ_{true}). The width of this distribution will allow us to estimate the uncertainty by quantifying the variance and the 1σ standard deviation from the distribution's mean. The next section, 4 will describe how varying different parameters within the data set can affect these uncertainty calculations. The second method is to plot the negative log likelihood versus the rate parameter λ and ascertain which value of λ minimizes the negative log likelihood curve. In order to calculate these values, the code takes each experiments estimated parameter and calculates a negative log likelihood result for the complete data set. This is very similar to the initial method of calculating a λ for each experiment but instead of estimating a lambda based off of the data in a single experiment, it is taking a lambda and computing its negative log likelihood result across the entire data set. A similar method with subtle differences. These two methods the lambda histogram and the negative log likelihood curve should give roughly the same result which will be outputted for comparison in a table by the script.

In addition to these two methods, since the Poisson function is a well-defined and well-behaved function, we can compare our maximum likelihood estimations to the analytical solution to the Poisson distribution. In this case, the most likely rate parameter is also equal to the average of the data points [9].

$$\lambda = \overline{X} = \left(\frac{\sum_{i=1}^{N_{meas}} X_i}{N_{meas}} \right) \tag{11}$$

As an example, for a data set with the parameters, $\lambda = 4$, $N_{exp} = 1000$, and $N_{meas} = 1000$, the analysis code with output the following plots Fig. 2 and tabulated values Fig. 3. The plots visually show the distribution of the data, the lambda histogram distribution, the negative log likelihood curve, and finally a Poisson distribution curve for the data given the true rate parameter. The table shows the computed values for the estimated rate parameter for the simulated data set for each aforementioned method (Lambda histogram, Negative log likelihood curve minimization, and finally the analytically derived average) as well as any associated variances or errors determined.

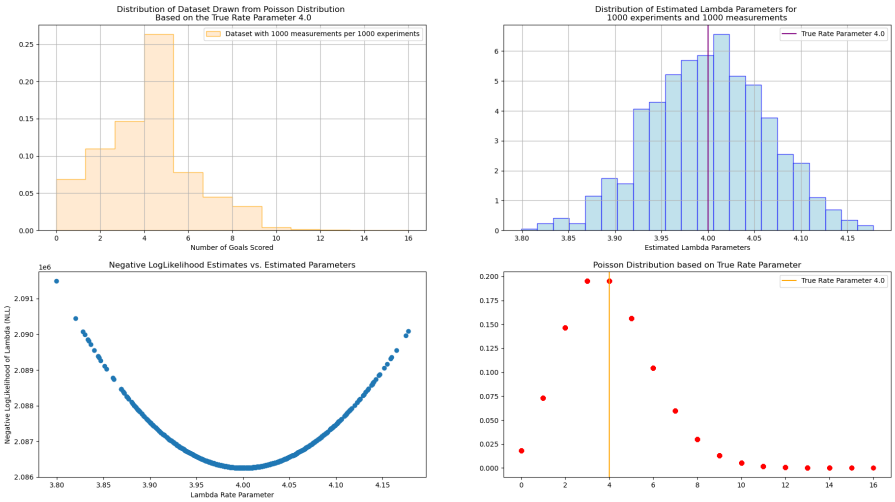

Figure 2: Output analysis plots for a simulated data set which was generated off of the parameters: rate $\lambda = 4$, $Nexp = 1000$, $Nmeas = 1000$. The top left panel shows a histogram of the data points. The top right panel shows a histogram of the estimated rate parameters which were calculated for each experiment. This histogram should peak at the true value. The bottom left panel shows a comparable plot of the parameter estimations versus the negative log likelihood values. The estimated rate parameter for the data set should be the λ that minimizes this curve. Lastly, the bottom right panel shows a depiction of the Poisson distribution for the data with the true rate indicated by a vertical line.

Description	Value
True Rate Parameter for Data	4.0
Number of Experiments	1000
Number of Measurments	1000
Lambda Histogram Estimated Mean:	4.000369212217579
Lambda Histogram Estimated Variance:	0.004018757633934757
Lambda Histogram Estimated StDev:	0.06339367187610098
Minimum of LogLikelihood Curve Estimate of Lambda:	4.000000028511696
Analytically derived Average:	4.000328
Analytically derived StDev:	2.000081998319069

Figure 3: A comparison table of the analysis calculations for a simulated data set which was generated off of the parameters: rate $\lambda = 4$, $Nexp = 1000$, $Nmeas = 1000$. The first section of this table shows characteristics of the data set ($\lambda, Nexp, Nmeas$). The following section shows the estimated mean (rate parameter), variance, and standard deviation for the histogram technique. The next section shows the rate parameter if we were to minimize the negative log likelihood curve. The final section shows the results compared to the analytically derived results since those can be computed for a Poisson distribution.

This concludes the analysis that is determined for each and every data set put into the MeasAnalysis.py script. The following section will discuss what trends and tests were conducted to analyze the behavior of different parameters within the data set and their affect on the analysis outputs.

4 Testing and Calculation of Detector Efficiency

Now that the simulation and analysis have been created, this section will discuss the generation and analysis of two different data sets. The only difference between the two data sets I analyzed is the detector efficiency. In one data set the detector efficiency was set to 0.50 and in the other it was set to 0.90. The other parameters, which we identical between the two data sets, are as follows: $N_{meas} = 1000$, $N_{exp} = 10,000$, $laserpower = 0.1$ Watts, and $Tmeas = 0.001$ attoseconds.

4.1 Analysis for Dataset with 90% Efficiency

In this test, the efficiency of the detector was set to 90%. With this value in hand, mean rate of photons per seconds was calculated by the code and this found the rate to be $\lambda = 452.76$ photons/second. The resulting data tables and plots are shown in Figures 4 and 5.

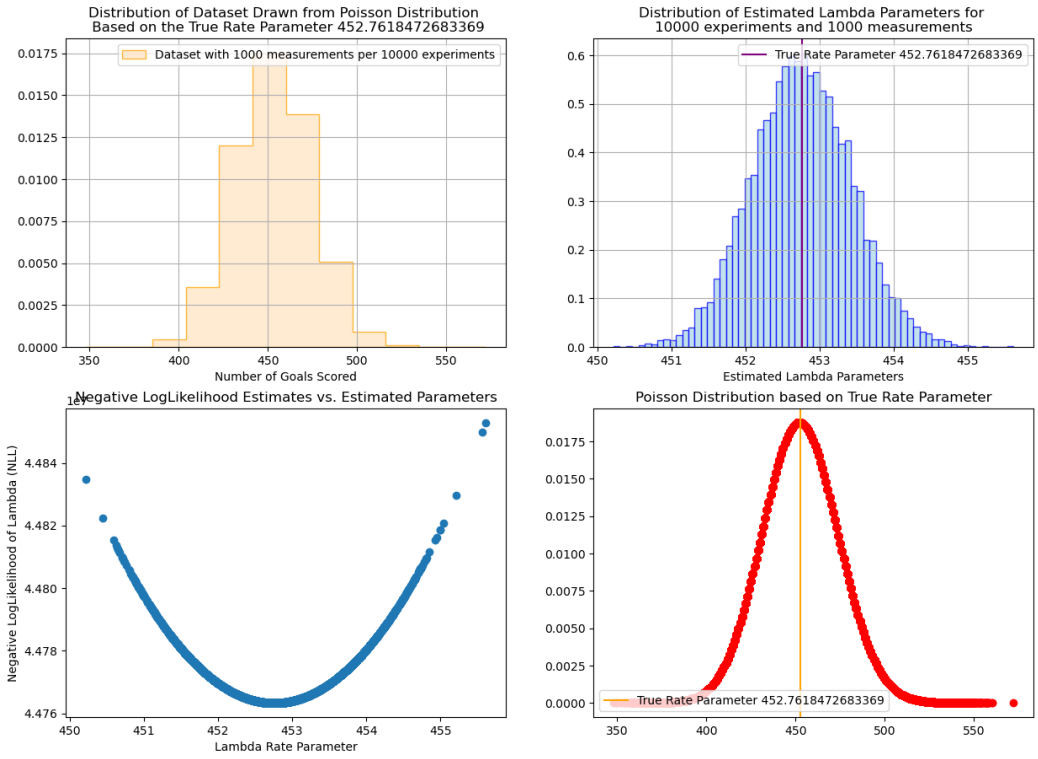

Figure 4: Output analysis plots for a simulated data set with 90% detector efficiency. The top left panel shows a histogram of the data points. The top right panel shows a histogram of the estimated rate parameters which were calculated for each experiment. This histogram should peak at the true value. The bottom left panel shows a comparable plot of the parameter estimations versus the negative log likelihood values. The estimated rate parameter for the data set should be the λ that minimizes this curve. Lastly, the bottom right panel shows a depiction of the Poisson distribution for the data with the true rate indicated by a vertical line.

Description	Value
True Rate Parameter for Data	452.7618472683369
Number of Experiments	10000
Number of Measurments	1000
Lambda Histogram Estimated Mean:	452.7563625399567
Lambda Histogram Estimated Variance:	0.450717771273034
Lambda Histogram Estimated StDev:	0.6713551752038812
Minimum of LogLikelihood Curve Estimate of Lambda:	452.7573392497487
Analytically derived Average:	452.7572699
Analytically derived StDev:	21.278093662262133

Figure 5: A comparison table of the analysis calculations for a simulated data set based on a 90% detector efficiency. The first section of this table shows characteristics of the data set ($\lambda, Nexp, Nmeas$). The following section shows the estimated mean (rate parameter), variance, and standard deviation for the histogram technique. The next section shows the rate parameter if we were to minimize the negative log likelihood curve. The final section shows the results compared to the analytically derived results since those can be computed for a Poisson distribution.

Based on this analysis, the code was able to estimate the rate of photons hitting the detector. If we input this into our equation from earlier we can calculate what the detector efficiency should be. The following equation shows how the detector efficiency (denote DE) can be calculated.

$$DE = \frac{nhc}{(laserpower)(T_{meas})} \tag{12}$$

where n is the mean number of photons per second (estimated by our analysis), h is Planck's constant, c is the speed of light constant, Tmeas is the time between measurements, and laserpower is the power of the laser being used in Watts.

When the value estimated by the analysis for the 90% efficiency data set, the resulting DE value was 0.90. A good match! This is a good indication that Ken would be able to use this analysis method along with his own experimental data to estimate the efficiency of his detector to relatively good degree of accuracy.

4.2 Analysis for Dataset with 50% Efficiency

In this test, the efficiency of the detector was set to 50%. With this value in hand, mean rate of photons per seconds was calculated by the code and this found the rate to be $\lambda = 251.53$ photons/second. The resulting data tables and plots are shown in Figures 6 and 7.

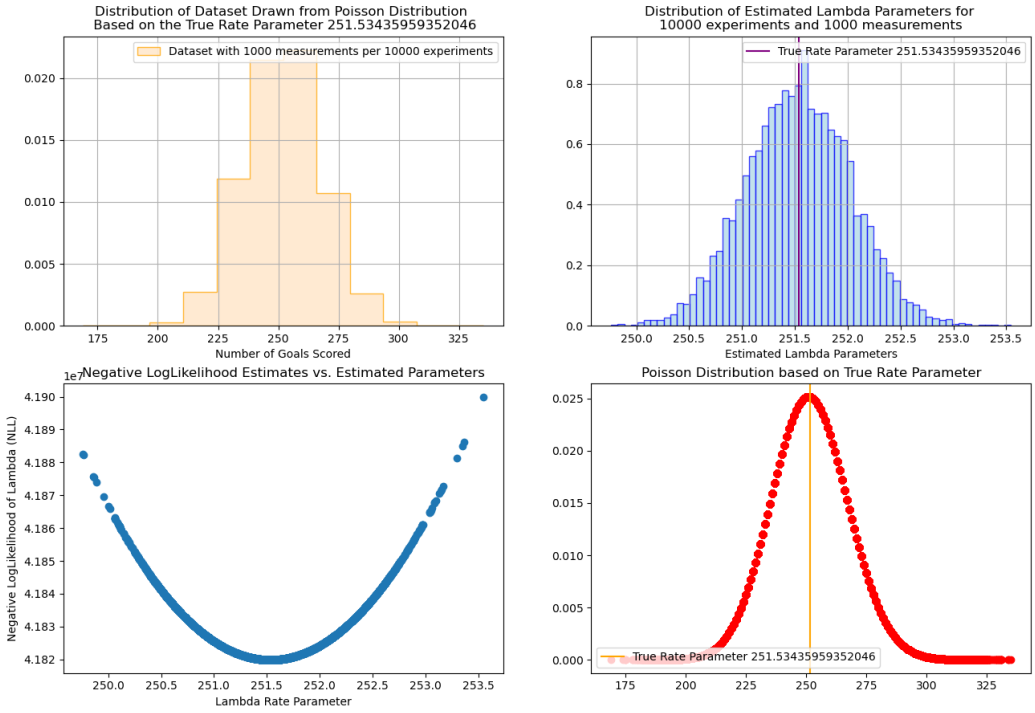

Figure 6: Output analysis plots for a simulated data set with 50% detector efficiency. The top left panel shows a histogram of the data points. The top right panel shows a histogram of the estimated rate parameters which were calculated for each experiment. This histogram should peak at the true value. The bottom left panel shows a comparable plot of the parameter estimations versus the negative log likelihood values. The estimated rate parameter for the data set should be the λ that minimizes this curve. Lastly, the bottom right panel shows a depiction of the Poisson distribution for the data with the true rate indicated by a vertical line.

Description	Value
True Rate Parameter for Data	251.53435959352046
Number of Experiments	10000
Number of Measurments	1000
Lambda Histogram Estimated Mean:	251.53323740309614
Lambda Histogram Estimated Variance:	0.24563802168458765
Lambda Histogram Estimated StDev:	0.4956188270078001
Minimum of LogLikelihood Curve Estimate of Lambda:	251.53383869553244
Analytically derived Average:	251.532917
Analytically derived StDev:	15.859789311337021

Figure 7: A comparison table of the analysis calculations for a simulated data set based on a 90% detector efficiency. The first section of this table shows characteristics of the data set ($\lambda, Nexp, Nmeas$). The following section shows the estimated mean (rate parameter), variance, and standard deviation for the histogram technique. The next section shows the rate parameter if we were to minimize the negative log likelihood curve. The final section shows the results compared to the analytically derived results since those can be computed for a Poisson distribution.

Once again, based on this analysis, the code was able to estimate the rate of photons hitting the detector. If we input this into our equation from earlier we can calculate what the detector efficiency should be. The following equation shows how the detector efficiency (denote DE) can be calculated.

$$DE = \frac{nhc}{(laserpower)(T_{meas})} \tag{13}$$

where n is the mean number of photons per second (estimated by our analysis), h is Planck's constant, c is the speed of light constant, Tmeas is the time between measurements, and laserpower is the power of the laser being used in Watts.

Similarly, when the value estimated by the analysis for the 50% efficiency data set, the resulting DE value was 0.90. A good match! This is a good indication that Ken would be able to use this analysis method along with his own experimental data to estimate the efficiency of his detector to relatively good degree of accuracy.

5 Conclusion

Overall, this project aimed to apply a real world physics scenario to the statistical simulations we have created throughout the course of this semester. I chose to follow an experiment based on photon counting from a laser modeling each photon as a count. This code allows for the simulation of photon counting data with the user inputting the parameters of their set up such as the power their laser, the interval of time between measurements, and the efficiency of their detector. These parameters allow for the calculation of the mean rate of photons that the detector should be receiving and recording per second. This was was then used to simulate data by sampling from a Poisson distribution that utilized that same rate of photons. This data can then be analyzed to estimate the mean rate of photons for the data. In order to perform this analysis, I employed the technique of maximum likelihood estimation, or equivalently, minimization of the negative log likelihood estimation. This allows for an estimation of the rate parameter to be made for each experiment. Beyond simply generating the data and analyzing each experiments most likely rate parameter, I also employed more techniques to estimate the mean for the entire data set. This was done by quantifying the spread of the parameter estimation data in a histogram. Additionally, a negative log likelihood curve was calculated and plotted for each data point across the range of potential rate parameters — whose minimum should equal the true rate parameter. Then based on the value that resulted for the overall, estimate rate parameter for the entire data set, the efficiency of the detector can then be calculated and compared to the true detector efficiency.

This code effectively demonstrates how an essential parameter of a model's distribution can be estimated from data alone which is an incredibly useful result for working with real world data since the true rate parameter would be unknown for a raw, observed data set. When considering the case of the scientist Ken and his own experimental setup, he would be able to utilize an analysis method such as the one presented in this paper to estimate the efficiency of his lab detector by using data collected by his set up. So even if he lost the instruction manual or specification sheet or never hears back from the manufacturer, Ken can rest assured that he can still estimate the efficiency of his detector to use in his later experiments and calculations.

6 Repository Link

GitHub Repository Link: `https://github.com/aelieber1/PHSX815_Project4`

References

[1] A. C. Sparavigna, *Poissonian distributions in physics: Counting electrons and photons*, Jan, 2021.

[2] GentecEO, , May, 2022.

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

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

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

[6] 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.

[7] *chapter 8: estimation of parameters and fitting of probability distributions*.

[8] B. Lindsey, *Understanding maximum likelihood estimation*, Nov, 2020.

[9] S. Towers, *Maximum likelihood estimation (MLE)*.