Optical Lab, Astronomy 120
Lab 1, Photon Counting and the Statistics of Light

Sameen Yunus
sfsyunus@berkeley.edu
Anthony Khodanian & Pavan Muddukrishna

09/19/2017

Abstract

In this lab, we studied the statistical properties of photon arrival times in a photomultiplier tube (PMT) using an LED of adjustable brightness. We studied the time intervals for 10000 samples recorded by the PMT. Our study utilized statistical methods that we learned through the course of the lab such as the standard deviation of the mean intervals between photons and fitting different probability density functions to our data. We learned the importance of integer types in data analysis with python and discovered atypical events known as "afterpulses" that skewed our data. Our results showed that uncertainty in data decreases with an increasing number of events. In addition, we showed that a histogram of lengths between photon arrivals follows an exponential curve and that the photon arrival rate for fixed time intervals follows a Poisson distribution.

Introduction

The photoelectric effect occurs when photons hit a metal or other material that is charged with electrons and excites the surface electrons into leaving the metal. This effect is described by Einstein's photoelectric equation[1] 01 which provides that the maximum kinetic energy of the photoelectrons must equal the difference in the energy of the absorbed photons and the work function.

$$eV_0 = h\nu - \Phi_0 \tag{01}$$

It follows that the photoelectric current I, the rate at which photoelectrons are emitted is proportional to N, the number of photons arriving per second. The PMT uses this effect to record the energy of the photon as a voltage. Our interest is in the arrival rate of photons measured in *clock ticks* which we analyzed in three different ways. We studied the statistical properties of the mean interval, looked at the length intervals between events and lastly we analyzed the photon arrival rate for evenly spaced time intervals.

[1] A derivation can be found on this website, *https://physics.info/photoelectric/*

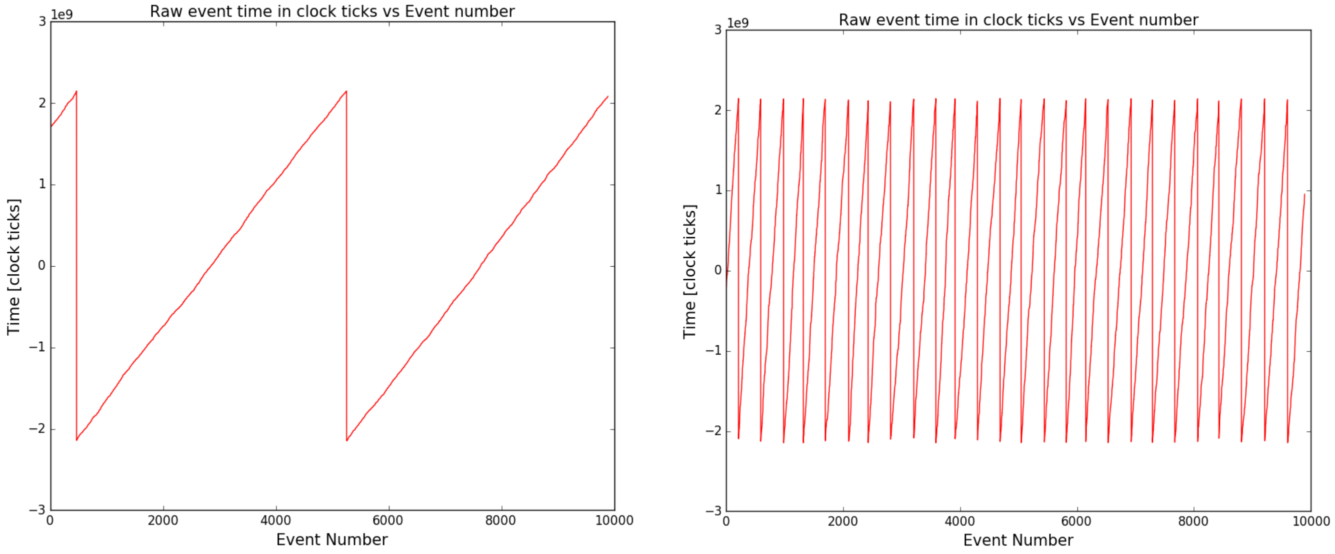
Data and Apparatus

This experiment uses a Hamamatsu H-6420-01 photomultiplier, an LED and a CoinPro which measures the time interval between pulses. Using the CoinPro GUI we recorded 10,000 photon events at maximum LED intensity. My group member, Anthony Khodanian used a data set with 1 million photon events, and as we learn later in this lab, the higher photon counts correspond to better precision so we should expect his plots to fit theoretical curves very well! The data set Pavan Muddukrishna and I used for our analyses was the 10,000 photon events recorded at maximum LED intensity. We also recorded time interval data for the LED at varying intensities between maximum and minimum (which was with the LED turned off). The experimental setup also contains a Tektronix oscilloscope and a squawker box. These allow us to actually observe the photon arrivals - on the oscilloscope, we can see the photon pulses for lower LED intensities with the time scale set to 50ns. The squawker amplifies the voltage pulses from the photons and translates them into clicks divided by time intervals corresponding to photon arrivals. We were able to hear the photon arrivals by changing the LED intensity and adjusting the squawker gain in order to hear individual clicks.

Data Analysis Methods

Data

Our raw data was composed of the photon arrival times measured in the CoinPro's *clock ticks* recorded for 10000 photons. Figure 1a below shows the raw data for the LED set to maximum brightness. This is the data set used for the majority of the analysis in this lab.



(a) *Raw time measured for 10000 samples with the LED at maximum intensity.*

(b) *Raw time measured for 10000 samples with the LED at minimum, non-zero intensity.*

Figure 1: *The sawtooth pattern in the graph comes from the the 32 bit limit of the CoinPro counter as it turns over from* 2^{31} *to* -2^{31}.

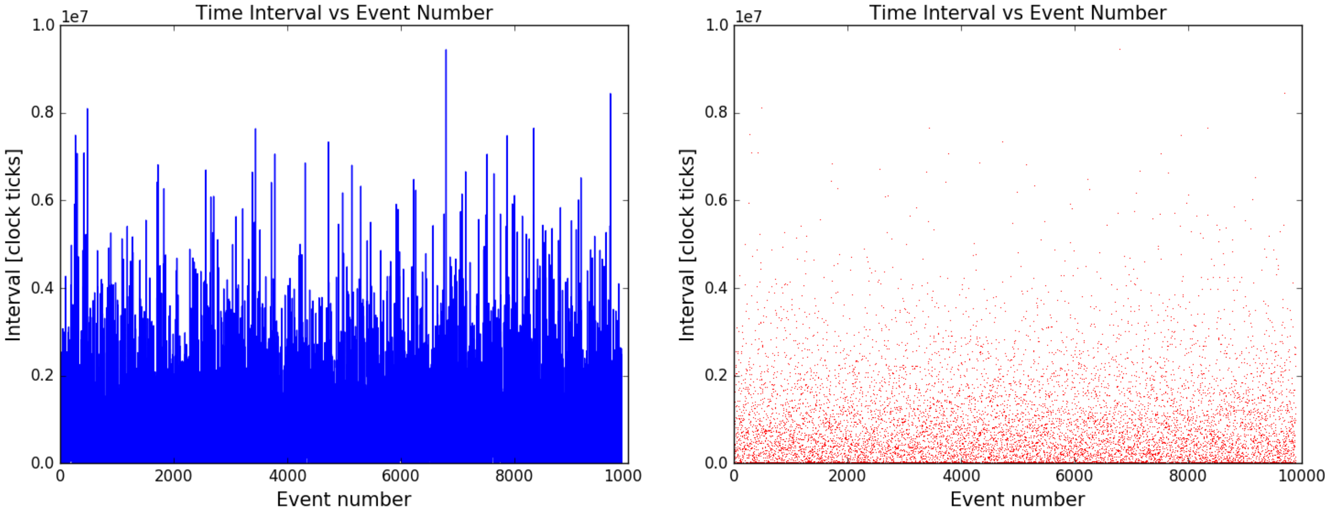
Note the sawtooth pattern in the raw data, our time values aren't actually negative. Since the computer has a 32 bit limit, when the clock tick value reaches its maximum at 2^{31} the computer rolls over to its minimum of -2^{31}. When we analyze the time intervals later, the program will know to use absolute values of clock ticks and we won't get any negative time intervals.

We use python numpy arrays to store and manipulate our data, numpy arrays are great for studies of statistical moments since it's very easy to perform mathematical operations on numpy arrays. It's important to set the datatype to 'int32' when we load our data file. If this is not done, python reads the numbers as integers instead of floating point numbers and this skews the data significantly when we study its statistical properties.

The sawtooth pattern depends on what intensity the LED is set at. Figure 1b shows the raw data for the LED set at a lower intensity. This seemed counter intuitive to me at first but the slope of the sawtooth represents the time passing between photon arrivals. A steeper slope means there's more time passing between photon events therefore the computer rolls over more times.

We compute the raw time intervals between photon arrivals by taking the difference of each subsequent time stamp by indexing the numpy array containing our timestamps. Figure 2a below shows us the roughly constant distribution of time intervals. We also see that the maximum time interval is about 9×10^7 clock ticks which is within the 32-bit range of the computer.

A secondary plot shows this same data distributed on a scatter plot, this allows us to see that there is a denser distribution of smaller time intervals. This implies that shorter time intervals, approaching 0 are more likely than longer ones.



(a) *Interval between subsequent events as a function of event number.*

(b) *Interval between subsequent events as a function of event number as a scatter plot.*

Figure 2: *We plot the lengths of the time intervals against the photon arrivals. Note in the scatter plot that most time intervals are quite short.*

Mean Interval and Standard Deviation

Once we had the length of the intervals, we computed the mean interval of these data. In order to do this we divided the data into chunks and calculated the sample mean of these chunks using Equation (01)[2] and then plotted these against the event index. We did this for 10 chunks of 1000 events each and also for 100 chunks of 100 events each; this is shown in the two graphs in Figure 3. We can see that the data starts to follow a more linear pattern as we divide it up into sample chunks.We can use a numpy method to compute the actual mean of the mean intervals to be, $\mu = 9.065 \times 10^5$ clock ticks. Looking at the plot for 100 chunks, we can see that the mean intervals lie more or less linearly around this line.

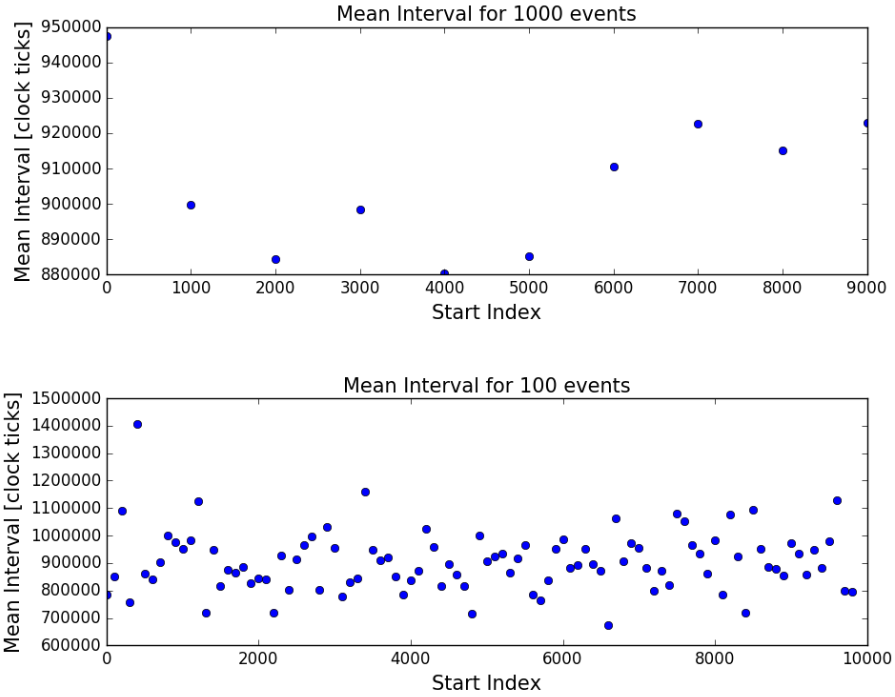


Figure 3: *The mean interval of the data computed for 10 evenly spaced chunks of 1000 samples and the mean interval for 100 evenly spaced chunks of 100 samples. Note that there are few deviations from the true mean of* 9.065×10^5 *clock ticks.*

As we can see, computing the mean by chunks brings the scattered distribution into an almost linear distribution. This suggests that as we average over a greater number of intervals, our mean should approach a constant value. This effect is observed in Figure 4 where we see the sample mean converge to the true mean, $\mu = 9.065 \times 10^5$.
In order to achieve this, we created a python for loop that averages the mean interval over progressively larger chunks where the chunk array was [100, 200, 300,...dt.size]. So the last data point is the average of the mean interval over all samples. In our case, the PMT recorded 9892 events initially and the number of intervals is then N = 9891. This proves to us the idea that a greater sample size increases the precision of data.

[2] This is equation 1 from the lab handout

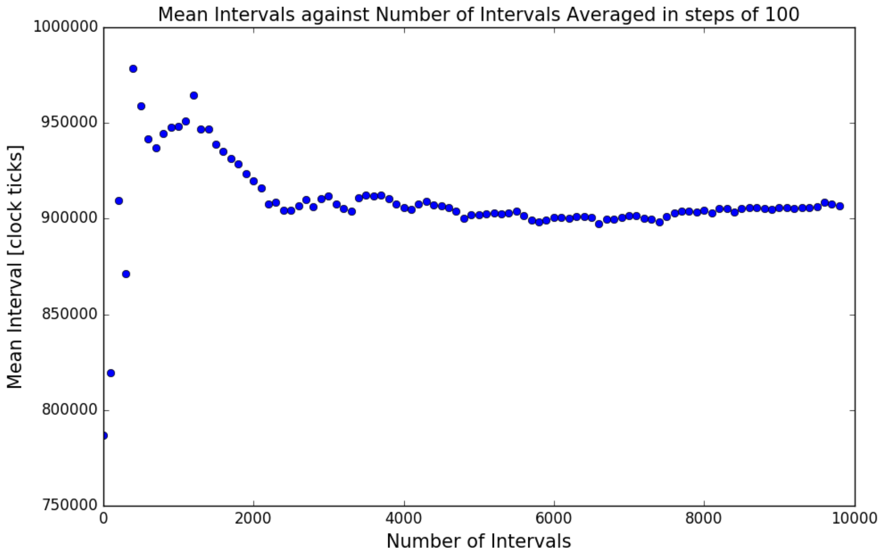
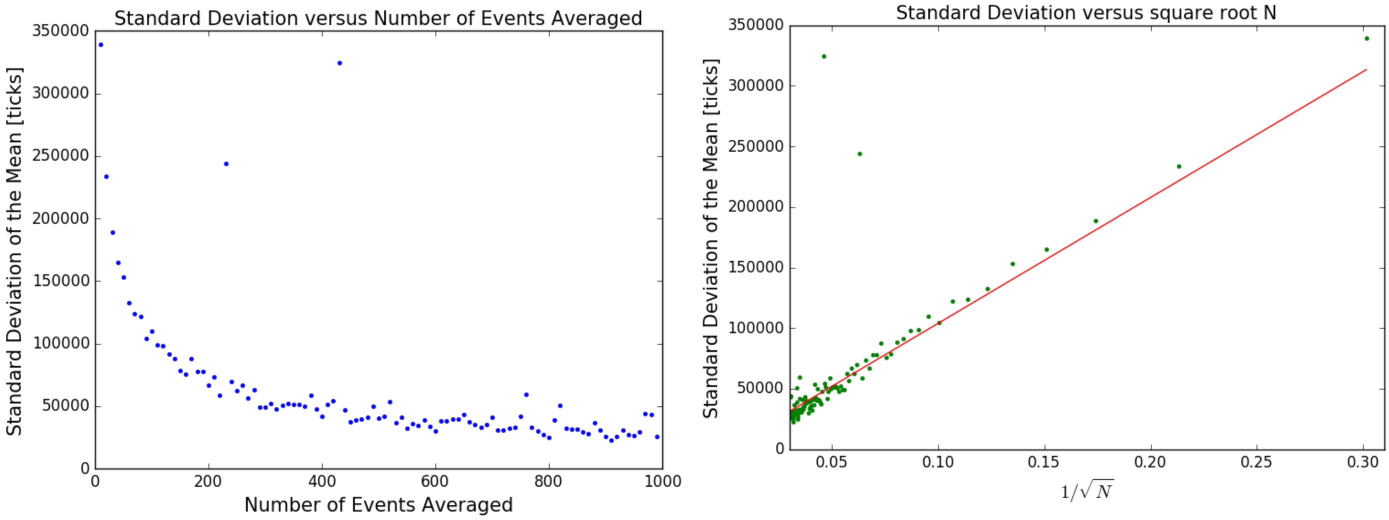


Figure 4: *The mean of data from 3 using progressively larger N, starting from N = 100 to N = 9891. Note that the mean approaches its true value as N increases.*

In order to study how our mean changes with increasing sample size, we make note of the behavior of the standard deviation of the means. We should expect to see that as N is increased, our mean comes closer to its true value and the standard deviation decreases. This is shown to be true in Figure 5a. We created a for loop that iterates over an array of N adding 10 each time so that N = [10, 20, 30,...,1000] and calculates the standard deviation.



(a) *Variation of the standard deviation of the mean decreases as we increase the N that we average over. Standard deviation describes the spread of the data, as N increases, so does our precision.*

(b) *Standard deviation of the mean vs. 1 / $\sqrt{\mathrm{N}}$ showing linear behavior. The red line is the theoretical expectation with SDOM = s / $\sqrt{\mathrm{N}}$, where s is the sample standard deviation.*

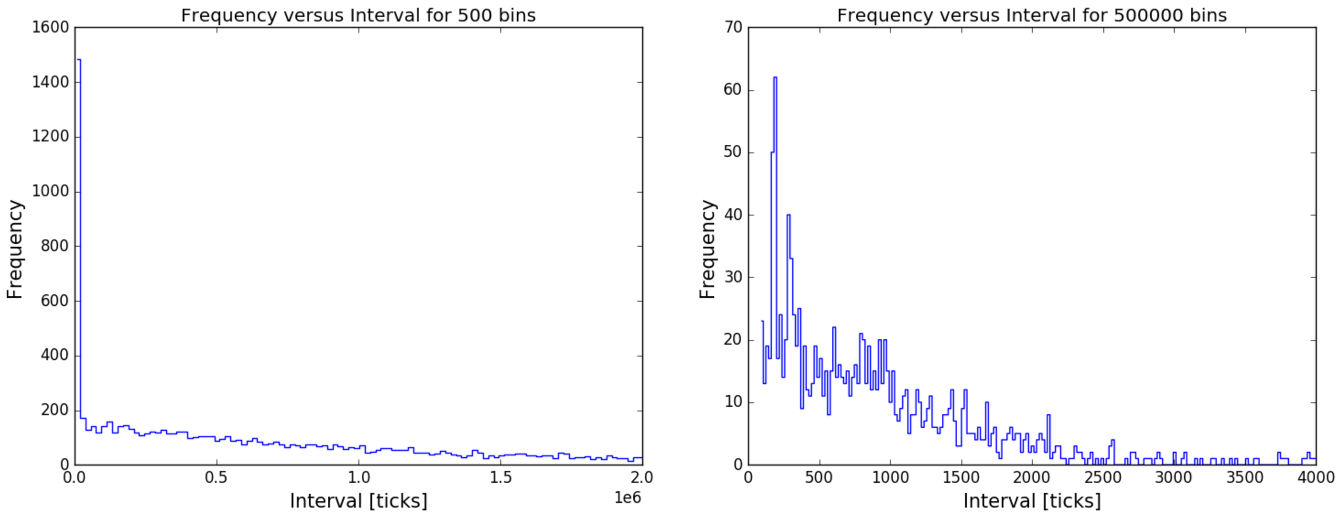
Figure 5: *Note how as we increase N, the standard deviation decreases as proof of our decreased uncertainty. The standard deviation scales with 1 / $\sqrt{\mathrm{N}}$ as we expect.*

The trend in Figure 5a indicates that the uncertainty in the data scales roughly with $1/\sqrt{\mathrm{N}}$ as the sample size N increases [3]. We show this to be true by graphing the standard deviation against $1/\sqrt{\mathrm{N}}$ and then plotting the straight line $s/\sqrt{\mathrm{N}}$ over this, where s is the standard deviation, $\sigma = 1.039 \times 10^6$ clock ticks.

One of the challenges we faced in this lab was figuring out the code to get the correct standard deviation of the mean intervals for chunks of N. Our group member Anthony Khodanian had the idea that the mean array had to be reinitialized after each iteration of N in order to get the correct standard deviations and that resulted in the plots we got.

Histograms

For this part of the analysis we created histograms by diving the time intervals into N = 500, evenly spaced bins of length Δt where the width of each bin is given by taking the maximum interval minus the minimum interval and dividing it into an even number of chunks that is defined by N. We then create an array, binl where each element is the the lowest edge of each subsequent bin and then loop over each bin and find how many events lie in each bin. This generates the histogram in Figure 6a. We can immediately see that most intervals are quite short and there appears to be an anomaly on the left side of the graph.



(a) *We count the number of photons arriving per bin for a fixed time interval. Note the histogram is skewed and clearly asymmetric.*

(b) *We inspect the spike to reveal a high density around 180 ticks (150ns) that is indicative of afterpulses that don't come from photons.*

Figure 6: *Histogram of the number of photons arriving for evenly spaced bins of length* Δt.

If we "Zoom in" on the histogram, so to speak, by increasing the number of bins to N = 500000, and looking only at the first 4000 intervals (this gets us a much better resolution) we can note the spike around 180 ticks that drops off up to about 3000 ticks. These

[3] This result actually follows from the Central Limit Theorem, "all of the samples will follow an approximate normal distribution pattern, with all variances being approximately..." $\sigma^2 = \frac{\sigma^2_{\text{population}}}{\mathrm{N}}$ *http://www.investopedia.com/terms/c/central_limit_theorem.asp#ixzz4szDPlYQA*

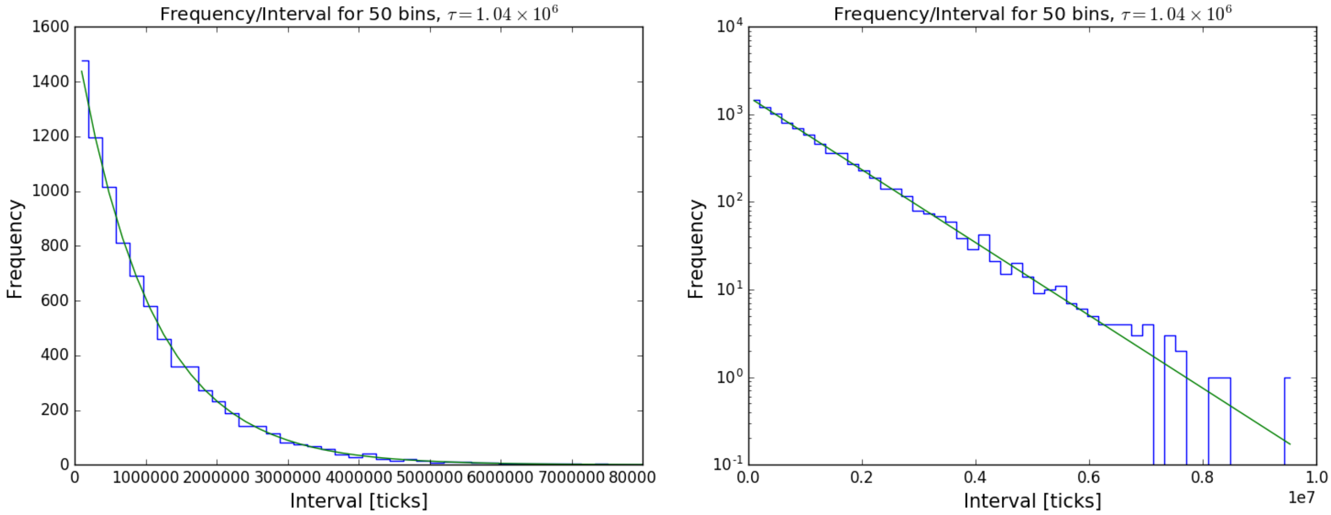
spurious events are known as *afterpulses* that result from ionization of remnant gas inside the PMT. These ions are collected at the photocathode and create many photoelectrons. They decrease resolution of data and make it harder to analyze low amplitude pulses[4]. In order to correct for this anomaly, we can "gate" our data of time intervals by only considering intervals that are longer 3000 clock ticks. We can perform a quick calculation to check how much of our data we lose this way to see if it falls under 15% as suggested by the lab handout.

$$\text{data}_{\text{lost}} = \frac{\text{dt.size} - \text{dt}_{\text{gated}}.\text{size}}{\text{dt.size}} \quad \rightarrow \quad \text{data}_{\text{lost}} = \frac{9891-8577}{9891} \quad \rightarrow \quad \text{data}_{\text{lost}} = 13.3\% \tag{02}$$

After removing afterpulses, we can model our gated data as an exponential curve following the equation;

$$p(\Delta t) = \frac{1}{\tau} e^{\frac{-\Delta t}{\tau}} \quad \Rightarrow \quad N = \frac{N_{\text{total}}}{\tau} e^{\frac{-\Delta t}{\tau}} \tag{03}$$

Where τ is the mean interval calculated from the gated data, Δt is the width of the bin, N_{total} is the sample size and N is the number of events per bin. We plot this theoretical exponential in both linear and log scale and see that it's an almost perfect fit in Figures 7a & 7b.



(a) *Note that our histogram fits an exponential curve quite well.*

(b) *To confirm the exponential fit, we plot on a logarithmic scale which results in a straight line.*

Figure 7: *Histogram of frequency of photons arriving per fixed length excluding intervals less than 3000 ticks with an exponential fit.*

This is a single parameter function where the mean and the standard deviation are both equal to the length of the bins; $\mu = \sigma = \Delta t$. We study this effect in the next section by looking at varying LED brightness.

[4] Please note, the description of afterpulses was written by me, Pavan and Anthony in conjunction after some research for our *Show and Tell*

Further Study of Exponential Distribution

As previously stated, for an exponential distribution $\mu = \sigma$ however it is hard to study this using one dataset. Therefore we collect six sets of 10000 samples at varying LED intensity. Anthony moved the intensity dial by six even turns between the maximum intensity and the LED being off therefore the resulting data is recorded at 100%, 80%, 60%, 40%, 20% and 0% intensities.
Having calculated the mean and standard deviation of each of these, we then plotted them against each other in Figure 8 and plotted a linear fit over these points. We can see they lie very close to a line described by y = x which shows the correlation between the mean and the standard deviation. The point closest to the origin is the maximum intensity point since the interval between photon arrivals is the smallest.

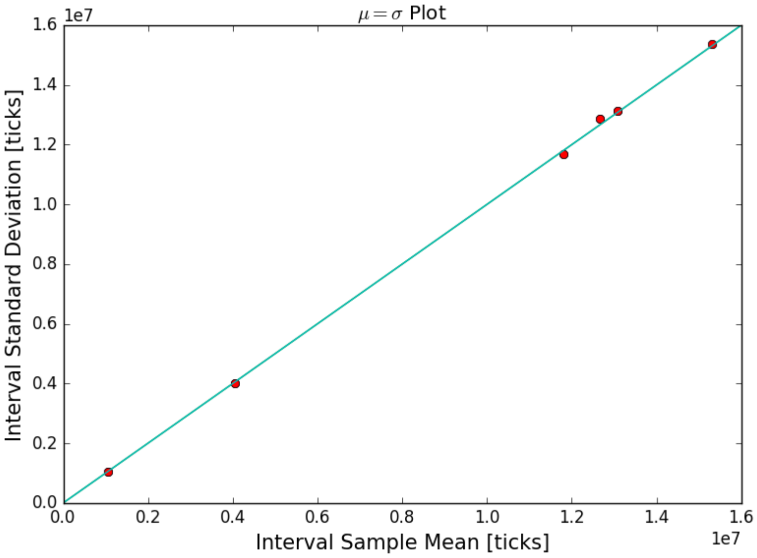


Figure 8: .

The Poisson Fit

In this section we show that our data can be described using a Poisson Distribution Function, however before we were able to create that fit we had to remove the 32-bit jumps from our original raw data in order to "stitch" the data together. To do this, there is a very useful numpy method that does the cumulative sum of all elements in an array. This plot is shown in 9.
We then plot the number of photons arriving per fixed time interval Δt for N = 2376 bins in the second graph in Figure 9. N is decided by the gated data that we have and our region of interest. Note that the distribution is more or less constant with time - if there were any systematic errors such as a voltage drift driving the LED, we would expect to see random fluctuations in the graph.
The second graph then loops over all the photons recorded in the first plot and bins them according to how many times a certain number of photons arrives per bin. We can fit a theoretical Poisson distribution to this plot.

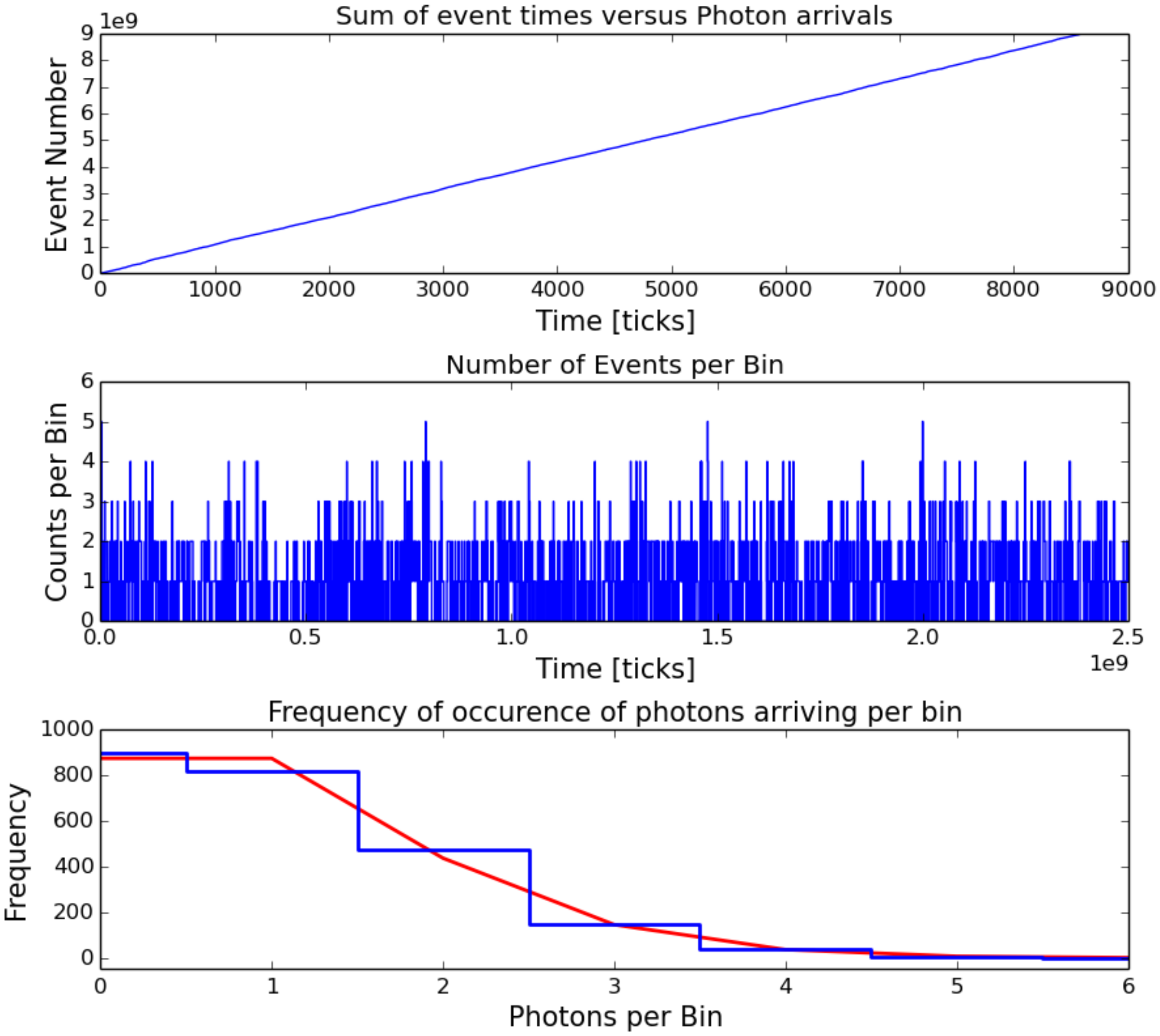


Figure 9: *The top graph shows the raw timestamps stitched together with the 32 bit roll over points removed.*
The second graph shows the number of photons arriving in N = 2376 evenly spaced bins.
The third graph plots the number of times a fixed number of photons arrived in a bin. Note that it follows a Poisson distribution that is plotted over it.

A Poisson distribution is described by the following Equation 04 when we have a given number of events occurring in a fixed time interval provided that;

a) the time at which each event occurs is independent of the time at which previous events occurred

b) the events occur at some known average rate, which in our case is the mean of the time intervals (Δt)

$$p(x;\mu) = \frac{\mu^x e^{-\mu}}{x!} \qquad \Rightarrow \qquad N = N_{total}\frac{\mu^x e^{-\mu}}{x!} \tag{04}$$

Where N is the number of times we count n photons arriving in a bin, N_{total} is the number of bins, x is the number of events per bin and μ is the mean number of photon arrivals.

Conclusion

Through the course of this lab, it became increasingly apparent that statistical analysis of data is what leads to actual interesting results. Raw data on its own is difficult to understand and typical data sets are too large to make estimations of trends based on plots of raw events. Our data analysis in this lab uncovered the that photon emissions follow an exponential distribution if we study the mean length of time intervals between photon arrivals and a poisson distribution if we study the rate of photon arrivals for fixed intervals. Going forward, we will be aware of the importance of measuring the uncertainties in our data, keeping in mind that a larger sample size will always improve random uncertainties. In addition, this lab emphasizes the importance collaborative work.