Poisson Distribution:
Counting Statistics*

Nicholas Munoz
University of Nevada, Las Vegas

(Dated: 9/17/21)

Abstract: In this experiment by observing the statistical measures based off of the observed data we can come to a conclusion using the chi square statistic and the ordinary least squares statistic to determine if the data sets follow a Poisson distribution, Gaussian distribution, or modeled no fit. The 1st and 4th data sets graphically followed a Poisson distribution while the 2nd data did not follow a Poisson distribution and modeled a Gaussian distribution. The 3rd collection of measurements we're more indicative of a Gaussian distribution.

Usage: Informational Purpose

I. INTRODUCTION

Cosmic background radiation is the remnant electromagnetic radiation from the primordial state of the earliest stages of the universe. The cosmic background radiation serves as key evidence surrounding numerous research studies that focus on developing leading theories on how the universe has developed in its early life to today. Cosmic background radiation (CMB) can be commonly found to be rich in the microwave region of the electromagnetic spectrum. Astronomers Arno Penzias and Robert Wilson coincidentally discovered the cosmic background radiation when they were mapping signals within Milk Way galaxy using a large horn antenna.

A Geiger counter is a measuring instrument that is able to be used to detect alpha particles, beta particles, Gamma radiation, and X-rays. Photons, are elementary particles that exhibit wave and particle like properties. Photons can be a resultant of radioactive decay and annihilation in which light is emitted in these processes.

In this experiment, we will be using the Geiger Counter to determine whether the rate at which particles are detected by the counter at set intervals of times of 0.1 minutes, 1 minute, and 10 minutes respectively, behave within the means of the Poisson and Gaussian statistical distributions. The process of measuring particles in this experiment is random with the expectation that any observation in equal amounts of time should not have the same outcome. The Gaussian distribution is expected with larger N sized samples while the Poisson distribution is to be expected with smaller sized samplings. The probability of occurrences for the measurements in this experiment will likely model one of the two distributions. The data taken should provide a conclusion to decide if it follows a Poisson distribution.

II. MEASUREMENTS AND PROCEDURES

The methods of collecting the counts for the data was obtained by initially deciding a voltage at which the Geiger counter would operate. This was done by first checking for the lower end voltage in which the counts began to count and then leading up to the upper end of the voltage right before it would exponentially increase. The upper and lower values of the voltage we're then averaged resulting in a voltage of 840 Volts.

To appropriately start collecting the data once the voltage was found would follow with working in time intervals of 0.1m, 1m, and 10 minutes. The fourth data set would be lead blocks surrounding the detector for the Geiger counter. It was important to choose sufficiently larger enough sample sizes of N = 250,100,20, and 250. Then the rest of the data was processed through using python and the respective mathematical functions to calculate the statistical measures and models of the data sets.

$$Mean : \Sigma \frac{x_i}{N}$$

$$Variance : \frac{1}{N-1} \Sigma (x_i - \bar{x})$$

$$Poisson : \frac{n^x}{e^n x!}$$

$$Gaussian : \frac{1}{\sigma \sqrt{(2\pi)}} e^{\frac{x-\mu}{2\sigma}} \frac{x-\mu}{2\sigma}$$

Both Poisson and Gaussian equations will be used to test if they are a good fit for taken measurements. The goodness of fit can be determined from the Chi Squared test

* Poisson Distribution: Counting Statistics

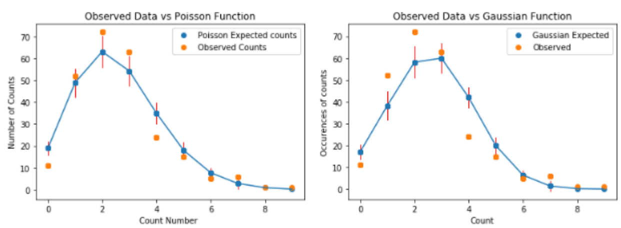
and Ordinary least Squares.

$$\chi^2 : \frac{1}{\sigma_i^2}\Sigma(x_i - \bar{x_i'})^2$$

$$OrdinaryLeastSquares : \frac{\chi^2}{N-1}$$

By comparing the Ordinary least squares to 1 we can determine whether the models are appropriate fits for the observed data. If it is equal to 1 then it is a model fit. If it is relatively bigger than 1 then it is likely to model what we are testing. If it is significantly larger than 1 than it will not be a good fit. And if it is less than 1 then there is noise and or an over fit that could be caused by the error of the variance and or a small sample size.

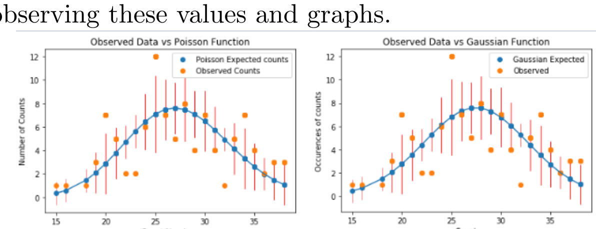
III. DATA ANALYSIS




```
Total Number of Counts:  250
Mean:  2.58
Standard Deviation:  1.5791055370648142
Variance:  2.493574297188752
Uncertainty of the Mean:  0.099871403258165
Uncertainty of the Variance:  0.2234794722179518
Degrees of Freedom: 9
Chi Square:  455.40931991216945
Ordinary Leasts Squares:  50.601035545796606
```

```
Total Number of Counts:  250
Mean:  2.58
Standard Deviation:  1.606237840420901
Variance:  2.58
Uncertainty of the Mean:  0.099871403258165
Uncertainty of the Variance:  0.2234794722179518
Degrees of Freedom: 8
Chi Square:  967.271886786631
Ordinary Leasts Squares:  120.90898584832887
```

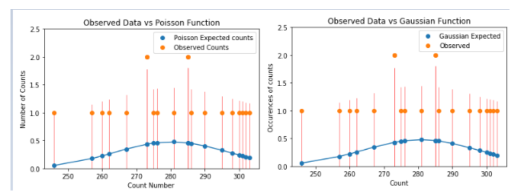
For data set one we conducted our experiment with a time interval of 0.1 minutes. This set was not shielded by lead castle. By using the chi squared test we end up with a ordinary least squares statistic of 50.6 for the Poisson distribution and 120.9 for the Gaussian distribution. Using these numbers we can determine that the Poisson distribution is more appropriately fit for this data because 120.9 is significantly larger than the other statistic relative to the value of 1. Thus, concluding that the Gaussian distribution is a poor fit and this set does follow a Poisson distribution by observing these values and graphs.




```
Total Number of Counts:  100
Mean:  27.48
Standard Deviation:  5.513033226231759
Variance:  30.393535353535356
Uncertainty of the Mean:  0.5513033226231758
Uncertainty of the Variance:  4.3199490066355395
Degrees of Freedom: 22
Chi Square:  130.53842693216558
Ordinary Leasts Squares:  5.9335648605529805
```

```
Total Number of Counts:  100
Mean:  27.48
Standard Deviation:  5.242136968832463
Variance:  27.48
Uncertainty of the Mean:  0.5513033226231758
Uncertainty of the Variance:  4.3199490066355395
Degrees of Freedom: 21
Chi Square:  133.8856714557672
Ordinary Leasts Squares:  6.375508164560343
```

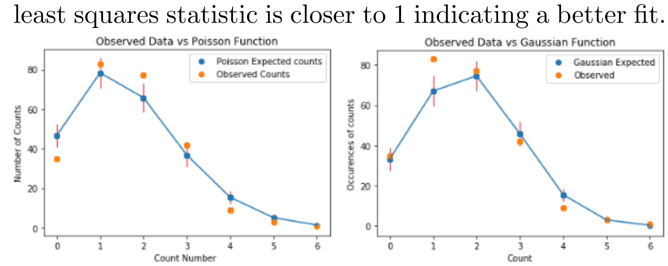
This data sets time interval was conducted a t = 1.0 minutes. Graphically, we can see that this data sets behaves with a more symmetric distribution. Poisson distributions are typically more appropriate for smaller sized data sets. Keeping this in mind, Poisson distributions will gradually follow Gaussian distributions as N increases. According to the Central Limit Theorem sample distributions will tend to a Gaussian with larger and larger sample sets having a normal distribution. Both models here could appropriately fit but we can see that it appropriate to say that this data set does not follow a Poisson distribution as well as it does a Gaussian distribution.




```
Total Number of Counts:  20
Mean:  280.75
Standard Deviation:  16.695019742741437
Variance:  278.7236842105263
Uncertainty of the Mean:  3.7331199030470903
Uncertainty of the Variance:  90.42990431086692
Degrees of Freedom: 17
Chi Square:  13.896701274148505
Ordinary Leasts Squares:  0.8174530161263827
```

```
Total Number of Counts:  20
Mean:  280.75
Standard Deviation:  16.75559608011604
Variance:  280.74999999999994
Uncertainty of the Mean:  3.7331199030470903
Uncertainty of the Variance:  90.42990431086692
Degrees of Freedom: 16
Chi Square:  13.885590082174172
Ordinary Leasts Squares:  0.8678493801358858
```

The time interval for this data set is t = 10.0 minutes. This data set has a sample size of N = 20. Because of this smaller sized sample set it was difficult to find a appropriate fit. Using the ordinary least squares both values are less than 1. When the ordinary least squares statistic is less than 1 it usually represents an over fit of data. When graphing the Gaussian without error bars in python the graph seemed more appropriately fit to a Gaussian. But since we are plotting the data along with the expected function the curvature of the function is flattened. It would be more likely to say that this data set is not following a Poisson distribution and closely follows a Gaussian distribution because the least squares statistic is closer to 1 indicating a better fit.




```
Total Number of Counts:  250
Mean:  1.68
Standard Deviation:  1.1308027500133484
Variance:  1.2787148594377487
Uncertainty of the Mean:  0.07151824548848354
Uncertainty of the Variance:  0.11460116597551266
Degrees of Freedom: 6
Chi Square:  453.1657120648359
Ordinary Leasts Squares:  75.52761867747265
```

```
Total Number of Counts:  250
Mean:  1.68
Standard Deviation:  1.296148139681572
Variance:  1.6799999999999997
Uncertainty of the Mean:  0.07151824548848354
Uncertainty of the Variance:  0.11460116597551266
Degrees of Freedom: 5
Chi Square:  454.25788646258087
Ordinary Leasts Squares:  90.85157729251617
```

This is the lead castle trial with time interval of t = 0.1 minutes. The effects of the lead lowered the expected number of counts as we see a decrease in the mean, number of bins(count numbers), and the accuracy of our numbers increased as there is a decrease in uncertainty of the mean and variance. Graphically, we can see that this data set is more appropriately fit to follow a Poisson distribution. The Chi-squared tests indicates that the Poisson distribution even though it is significantly larger than 1, is a better fit than the Gaussian model, leading to the conclusion that this data follows a Poisson distribution.

IV. CONCLUSION

After doing graphical and statistical analysis, the data sets certainly do model either a Poisson distribution or a Gaussian distribution. In some cases it is possible based off of the observed data that neither models fit our data. It is expected that Poisson distributions will eventually normalize over an arbitrarily large value of N due to the Central Limit Theorem. The count rate did vary significantly between the time intervals and the leaded castle as not all sets exhibited the same model behavior.

[1] Alaina G. Levine *: The Large Horn Antenna and the Discovery of Cosmic Microwave Background Radiation. https://www.aps.org. Date Accessed: 9/16/2021*

Physics-Code (/github/Raiziel/Physics-Code/tree/master)
/ Poisson distribution.ipynb (/github/Raiziel/Physics-Code/tree/master/Poisson distribution.ipynb)

In [1]:

```
#for i in rset:

      #  Pset2z = N*np.array(Pset2)
       # print ("P(r,λ) for r value of", i, "is",Pset2z[i])
#plt.plot(rset,Pset2z)
#plt.title("Distribution of Probability for Each R Count")
#plt.xlabel("r ")
#plt.ylabel("P(r,λ)")
```

In [2]:

```
import statistics
import numpy
import math
import numpy as np
from numpy import log as ln
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt
from scipy.optimize import minimize
from scipy.integrate import quad
from scipy.stats import chisquare
from scipy.special import gamma

from decimal import Decimal

# 0.1m no shield
DataSet1 = [1,2,3,0,4,2,1,5,5,3,2,4,3,4,5,5,1,2,2,3,1,2,1,5,2,4,1,3,3,3,3,3,
          ,1,2,3,1,7,2,3,3,2,2,1,2,2,2,3,1,1,0,3,3,0,3,7,3,4,2,2,3,4,6,2,3,4
          ,2,2,2,2,2,3,3,7,5,3,7,4,0,3,1,4,3,4,4,3,2,1,1,4,3,2,1,1,2
          ,3,2,2,1,1,2,1,7,2,3,3,3,1,2,1,3,2,5,2,1,4,5,1,1,1,1,4,1,1,2,2,2,5
          ,2,2,2,1,2,0,5,3,3,5,1,4,3,2,0,3,1,4,2,3,3,3,5,2,2,7,4,1,3,3,2,2
          ,3,2,2,0,2,0,5,2,3,4,2,3,3,2,2,3,2,4,3,8,3,2,2,0,2,3,4,3,6,1,1,2,6
          ,2,2,1,6,3,2,1,0,2,1,2,2,1,3,2,1,3,2,3,3,2,3,3,1,2,1,4,1,1,0,2,1,2
# 1m
DataSet2 = [31,28,34,25,38,37,37,31,38,30,22,36,25,28,33,35,26,24,34,33,26,2
            ,23,22,36,19,27,35,37,20,20,34,20,31,30,21,27,29,28,30,35,33,25,1
            ,19,28,30,24,25,35,27,25,33,38,16,20,25,21,34,26,25,26,25,21,20,2
            ,26,34,20,25,26,24,29,19,27,34,24,21,32,28,25,29,30,28,30,24,18,2

# 10m
DataSet3 = [285,298,260,273,285,262,276,281,286,290,273,303,257,267,275,295,
# 0.1m w/ Shield
DataSet4 = [2,0,1,0,2,2,4,1,3,0,2,0,2,2,2,1,1,1,2,0,1,1,1,2,2
            ,0,2,2,2,3,2,3,2,2,2,4,2,3,1,2,4,1,2,2,1,4,2,1,1,4
            ,2,1,1,0,4,1,1,3,3,3,1,0,1,1,1,1,3,2,3,2,1,1,2,3,2
            ,3,2,1,5,0,3,2,1,2,2,2,3,6,2,3,1,2,2,2,2,2,2,1,0,3
            ,3,2,1,3,0,2,1,0,1,1,0,2,0,2,2,2,3,1,1,2,0,0,5,1,1
            ,2,1,3,2,1,3,1,2,2,2,4,0,2,0,1,0,1,3,1,1,2,2,1,1,3
            ,3,0,3,1,1,2,3,1,3,0,2,0,2,0,1,1,1,5,1,3,2,0,0,3,1
            ,1,3,0,1,2,3,2,1,3,1,0,3,2,2,1,3,1,0,1,3,2,3,0,1,1
            ,4,2,2,2,1,2,1,2,3,1,0,1,1,3,1,2,0,1,2,2,0,1,0,1,1
            ,3,2,1,1,1,3,3,2,0,1,2,3,1,3,2,2,1,1,1,1,1,1,2,4,0]



DataSet1.sort()
DataSet2.sort()
DataSet3.sort()
DataSet4.sort()
```

In [3]:

```
def SampleSize(Set):
    N = len(Set)
    return N
```

In [4]:

```
# function to calculate factorial
def Factorial(N):

    factorial = 1
    if int(N) >= 0:
        for i in range (1,int(N)+1):
                factorial = factorial * i
        return factorial
```

In [48]:

```
# function to plot histogram
def Histogram(Set,Stat):

    mean,STD,variance,uncertainty=Stat
    N = SampleSize(Set)

    num_bins = N
    N, bins, patches = plt.hist(Set, num_bins, facecolor='blue',edgecolor='b
    plt.title("Rate of occurence of counterate")
    plt.xlabel("count")
    plt.ylabel("Frequency of Occurence")
    plt.subplots_adjust(left=0.15)
    plt.show()
```

In [6]:

```
def Stat(Set):
    N = SampleSize(Set)
    mean = (statistics.mean(Set))
    STD = (statistics.stdev(Set,xbar=None))
    variance = (statistics.variance(Set))
    Uncertainty = (STD/(math.sqrt(N)))

    return mean,STD,variance,Uncertainty
```

In [57]:

```
def ChiSquared(Stat,Factorial,Set,SampleSize,Count):
    Pset=[]
    Eset=[]
    Probability=[]
    Cset = Count
    mean,STD,variance,uncertainty=Stat
    N=SampleSize

    for i in Set:
        if(i>100):
            P = (((np.exp(i-mean))*((mean/i)**i))/ (math.sqrt(2*np.pi*i)))*N
            #print ("P for r value of", i, "is", P)
            Pset.append(P)

        else:
            P = (((mean**i)*(math.exp(-mean)/(Factorial(i)))))*N
            #print ("P for r value of", i, "is", P)
            Pset.append(P)

    Pstd = statistics.stdev(Pset)
    PUncertain = (Pstd/(math.sqrt(N)))

    for i in Cset:
        prob = (i/N)
        Probability=np.append(Probability,prob)
    for i in Probability:
        Estat = np.sqrt(N*i*(1-i))
        Eset=np.append(Eset,Estat)
    ChiSquare =[]
    ChiSum = 0
    for i in range(len(Set)):
        ChiStat = np.square((Cset[i]- Pset[i]))/np.square(Eset[i])
        ChiSum = ChiStat+ChiSum
        ChiSquare=np.append(ChiSquare,ChiStat)

    #print(ChiSquare)
    return ChiSum
    #print(len(Set))
```

In [174]:

```
# function to model the occurence of counts of particles hitting the detectc
def Poisson(Stat,Factorial,Set,SampleSize,Count):
    Pset=[]
    Eset=[]
    Probability=[]
    Cset = Count

    mean,STD,variance,uncertainty=Stat
    N=SampleSize

    for i in Set:
        if(i>100):
            P = (((np.exp(i-mean))*((mean/i)**i))/ (math.sqrt(2*np.pi*i)))*N
            #print ("P for r value of", i, "is", P)
            Pset.append(P)

        else:
            P = (((mean**i)*(math.exp(-mean)/(Factorial(i)))))*N
            #print ("P for r value of", i, "is", P)
            Pset.append(P)

    Pstd = statistics.stdev(Pset)
    PUncertain = (Pstd/(math.sqrt(N)))

    for i in Cset:
        prob = (i/N)
        Probability=np.append(Probability,prob)
    for i in Probability:
        Estat = np.sqrt(N*i*(1-i))
        Eset=np.append(Eset,Estat)




    a,b,c,d,e = Set,Pset,Cset,Probability,Eset

    plt.plot(Set,Pset,'o',label='Poisson Expected counts')
    plt.plot(Set,Cset,'o',label='Observed Counts')
    plt.errorbar(a, b, e, xerr=None, fmt='', ecolor='r', elinewidth=0.5, car
    plt.xlabel("Count Number ")
    plt.ylabel("Number of Counts")
    plt.title("Observed Data vs Poisson Function")
    plt.legend()
```

In [9]:

```
def Count(Set):
    count=[]

    for i in Set:
        Z=Set.count(i)
        #print("Count R of ", i, "is", Z)
        count.append(Z)
    return count
```

In [10]:

```
#Keep
def Variance(DataSet):
    N = len(DataSet)
    VarSum = 0
    x = 1/(N-1)
    for i in DataSet:
        VarSum = VarSum + (i - statistics.mean(DataSet))**2

    return x*VarSum
```

In [11]:

```
# Keep
def ErrorMean(DataSet):
    N = len(DataSet)
    VarSum = 0
    x = 1/(N-1)
    for i in DataSet:
        VarSum = VarSum + (i - statistics.mean(DataSet))**2

    Ssquare = x*VarSum
    EM = np.sqrt(Ssquare)/np.sqrt(N)

    return EM
```

In [167]:

```
from scipy.stats import norm
def Gaussian(DataSet):
    k=len(set(list(DataSet)))-2
    N=len(DataSet)
    m = statistics.mean(DataSet)
    std=np.sqrt(m)
    Evar=std**2*np.sqrt(2)/(np.sqrt(N-1))
    Gset=[]
    Cset=Count(DataSet)
    Probability=[]
    Eset=[]
    Gset=[]
    L = 1/(std*np.sqrt(2*np.pi))
    B = 2*(np.square(std))
    for i in DataSet:
        G = N*L*np.exp(-((np.square(i- m))/(B)))
        Gset.append(G)


    for i in Cset:
        prob = (i/N)
        Probability=np.append(Probability,prob)

    for i in Probability:
        Estat = np.sqrt(N*i*(1-i))
        Eset=np.append(Eset,Estat)
    ChiSquare =[]
    ChiSum = 0
    for i in range(len(DataSet)):
        ChiStat = np.square((Cset[i]- Gset[i]))/np.square(Eset[i])
        ChiSum = ChiStat+ChiSum
        ChiSquare=np.append(ChiSquare,ChiStat)

    a,b,c,d,e = DataSet,Gset,Cset,Probability,Eset


    plt.plot(DataSet,Gset,'o',label="Gaussian Expected")
    plt.xlabel("Count")
    plt.ylabel("Occurences of counts")
    plt.title("Observed Data vs Gaussian Function")
    plt.plot(DataSet,Cset,'o',label="Observed")
    plt.errorbar(a, b, e, xerr=None, fmt='', ecolor='r', elinewidth=0.5, cap
    plt.legend()

    print("Total Number of Counts: ", N)
    print("Mean: ", statistics.mean(DataSet))
    print("Standard Deviation: ", std)
    print("Variance: ", std**2)
    print("Uncertainty of the Mean: ",ErrorMean(DataSet))
    print("Uncertainty of the Variance: ",ErrorVar(DataSet))
    print("Degrees of Freedom:", k)
    print("Chi Square: ",ChiSum)
    print("Ordinary Leasts Squares: ",ChiSum/k)
```

In [12]:

```
# KEEP
def ErrorVar(DataSet):
    N = len(DataSet)
    VarSum = 0
    x = 1/(N-1)
    for i in DataSet:
        VarSum = VarSum + (i - statistics.mean(DataSet))**2

    Ssquare = x*VarSum
    S = np.sqrt(Ssquare)/np.sqrt(N)


    EV = Ssquare*np.sqrt(2)/(np.sqrt(N-1))

    return EV
```

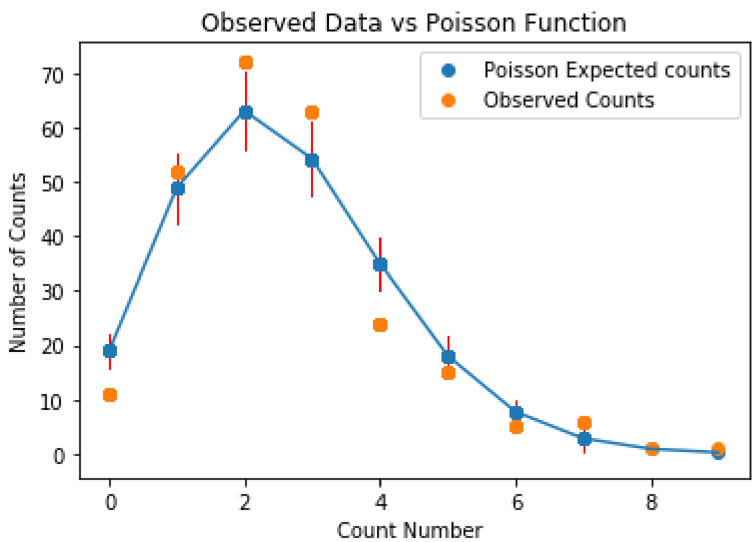
In [13]:

```
# KEEP
def Error(DataSet):
    N = len(DataSet)
    k=len(set(list(DataSet)))-1
    #Sample Standard Deviation
    Std = statistics.stdev(DataSet)
    Chi=ChiSquared(Stat(DataSet),Factorial,DataSet,SampleSize(DataSet),Count
    #Chi Square
    #ChiSquare = 0
    #for i in DataSet:
        #ChiSquare = ChiSquare + np.square((i-statistics.mean(DataSet)/Svar)
        #print (ChiSquare, i)
    print("Total Number of Counts: ", N)
    print("Mean: ", statistics.mean(DataSet))
    print("Standard Deviation: ", Std)
    print("Variance: ", Variance(DataSet))
    print("Uncertainty of the Mean: ",ErrorMean(DataSet))
    print("Uncertainty of the Variance: ",ErrorVar(DataSet))
    print("Degrees of Freedom:", len(set(list(DataSet)))-1)
    print("Chi Square: ",Chi)
    print("Ordinary Leasts Squares: ",Chi/k)
    #print("Chi Square Distribution Value: ",ChiSquare)
```

In [175]:

```
# Data Set 1 : t = 1m, N = 120, V = 800


Poisson(Stat(DataSet1),Factorial,DataSet1,SampleSize(DataSet1),Count(DataSet
#Histogram(DataSet1,Stat(DataSet1))
Error(DataSet1)
```

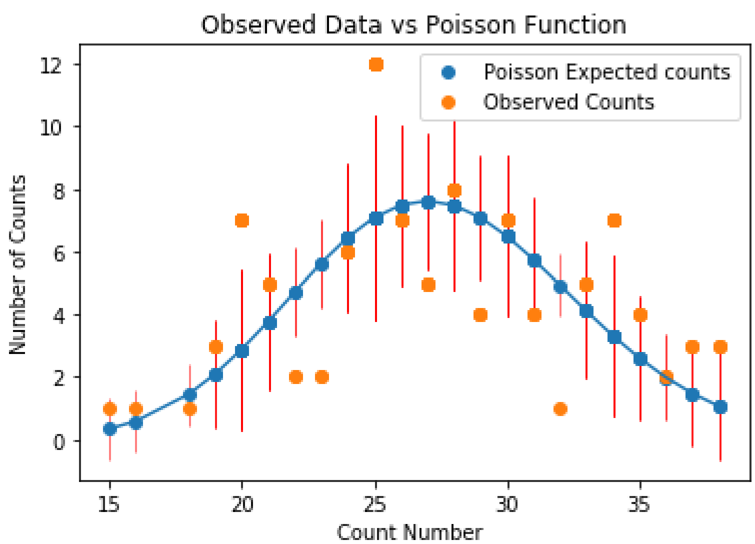
```
Total Number of Counts:  250
Mean:  2.58
Standard Deviation:  1.5791055370648142
Variance:  2.493574297188752
Uncertainty of the Mean:  0.099871403258165
Uncertainty of the Variance:  0.2234794722179518
Degrees of Freedom: 9
Chi Square:  455.40931991216945
Ordinary Leasts Squares:  50.601035545796606
```



In [176]:

```
# Data Set 2 : t = 1m, N = 100, V = 840

SampleSize(DataSet2)
Stat(DataSet2)
Poisson(Stat(DataSet2),Factorial,DataSet2,SampleSize(DataSet2),Count(DataSet
#Histogram(DataSet2,Stat(DataSet2))
Error(DataSet2)
```

```
Total Number of Counts:  100
Mean:  27.48
Standard Deviation:  5.513033226231759
Variance:  30.393535353535356
Uncertainty of the Mean:  0.5513033226231758
Uncertainty of the Variance:  4.3199490066355395
Degrees of Freedom: 22
Chi Square:  130.53842693216558
Ordinary Leasts Squares:  5.9335648605529805
```



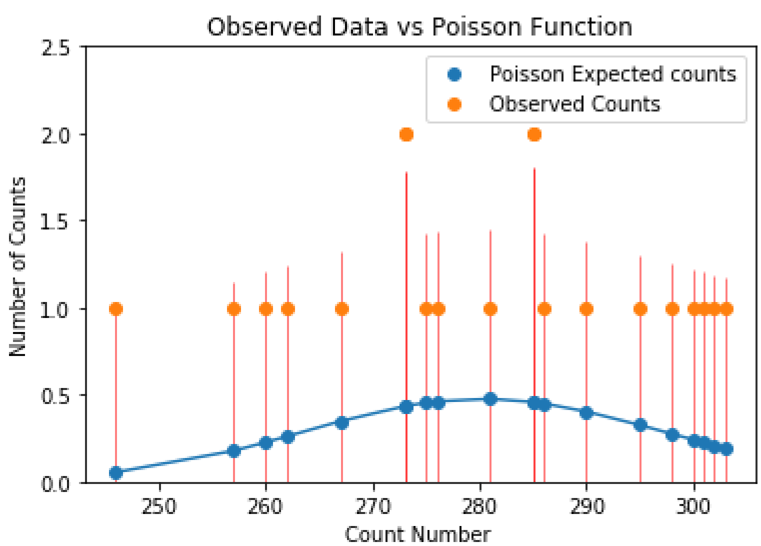
In [184]:

```
# Data Set 3 : t = 10.0m, N = 20, V = 840



SampleSize(DataSet3)
Stat(DataSet3)
Poisson(Stat(DataSet3),Factorial,DataSet3,SampleSize(DataSet3),Count(DataSet
#Histogram(DataSet3,Stat(DataSet3))
Error(DataSet3)
plt.ylim([0,2.5])
```

```
Total Number of Counts:  20
Mean:  280.75
Standard Deviation:  16.695019742741437
Variance:  278.7236842105263
Uncertainty of the Mean:  3.7331199030470903
Uncertainty of the Variance:  90.42990431086692
Degrees of Freedom: 17
Chi Square:  13.896701274148505
Ordinary Leasts Squares:  0.8174530161263827
```

Out[184]:

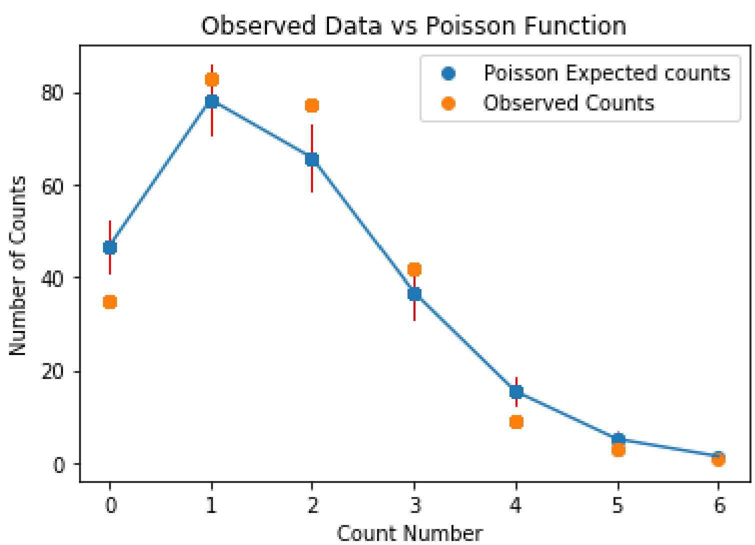
```
(0, 2.5)
```



In [178]:

```
# Data Set 4 : t = 0.1m, N = 250, V = 840



SampleSize(DataSet4)
Stat(DataSet4)
Poisson(Stat(DataSet4),Factorial,DataSet4,SampleSize(DataSet4),Count(DataSet
#Histogram(DataSet4,Stat(DataSet4))
Error(DataSet4)
```

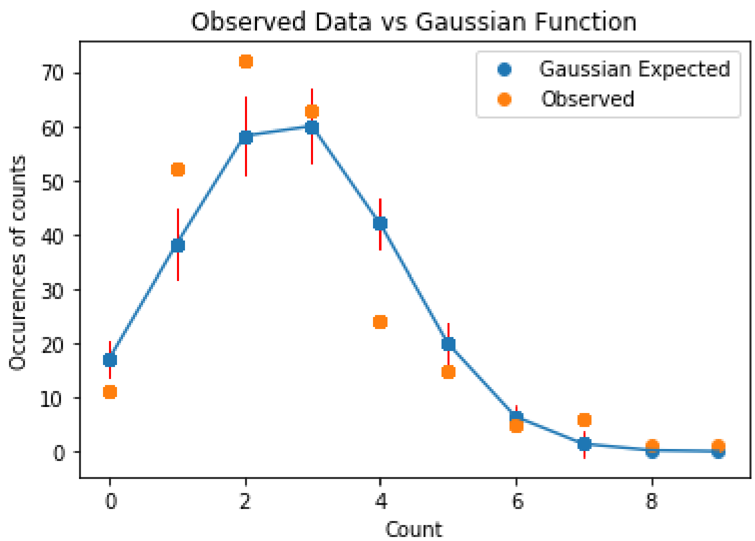
```
Total Number of Counts:  250
Mean:  1.68
Standard Deviation:  1.1308027500133484
Variance:  1.2787148594377487
Uncertainty of the Mean:  0.07151824548848354
Uncertainty of the Variance:  0.11460116597551266
Degrees of Freedom: 6
Chi Square:  453.1657120648359
Ordinary Leasts Squares:  75.52761867747265
```



In [168]:

```
Gaussian(DataSet1)
```

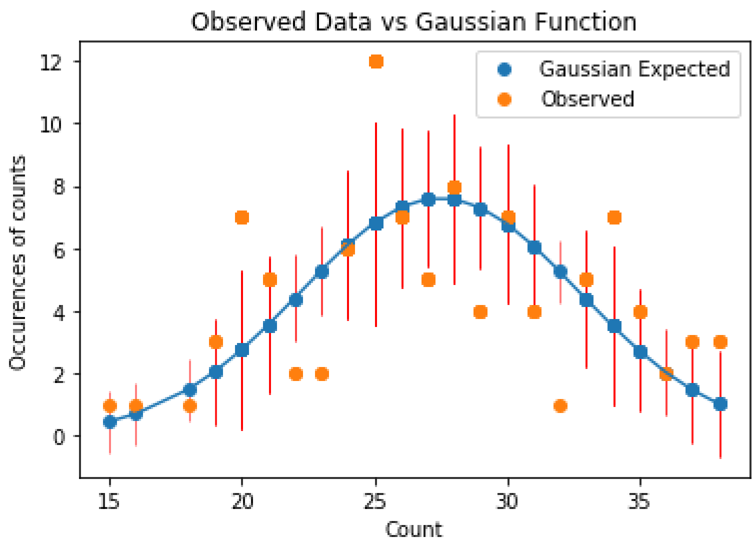
```
Total Number of Counts:  250
Mean:  2.58
Standard Deviation:  1.606237840420901
Variance:  2.58
Uncertainty of the Mean:  0.099871403258165
Uncertainty of the Variance:  0.2234794722179518
Degrees of Freedom: 8
Chi Square:  967.271886786631
Ordinary Leasts Squares:  120.90898584832887
```



In [169]:

```
Gaussian(DataSet2)
```

```
Total Number of Counts:  100
Mean:  27.48
Standard Deviation:  5.242136968832463
Variance:  27.48
Uncertainty of the Mean:  0.5513033226231758
Uncertainty of the Variance:  4.3199490066355395
Degrees of Freedom: 21
Chi Square:  133.8856714557672
Ordinary Leasts Squares:  6.375508164560343
```



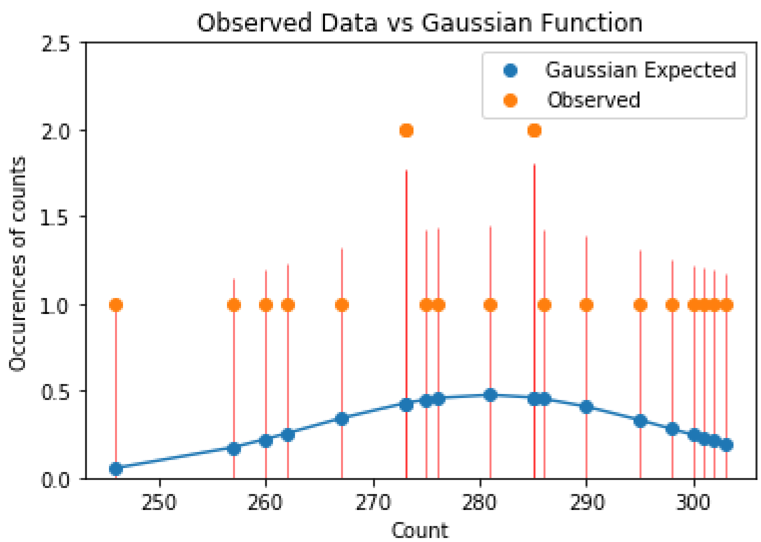
In [183]:

```
Gaussian(DataSet3)
plt.ylim([0,2.5])
```

```
Total Number of Counts:  20
Mean:  280.75
Standard Deviation:  16.75559608011604
Variance:  280.74999999999994
Uncertainty of the Mean:  3.7331199030470903
Uncertainty of the Variance:  90.42990431086692
Degrees of Freedom: 16
Chi Square:  13.885590082174172
Ordinary Leasts Squares:  0.8678493801358858
```

Out[183]:

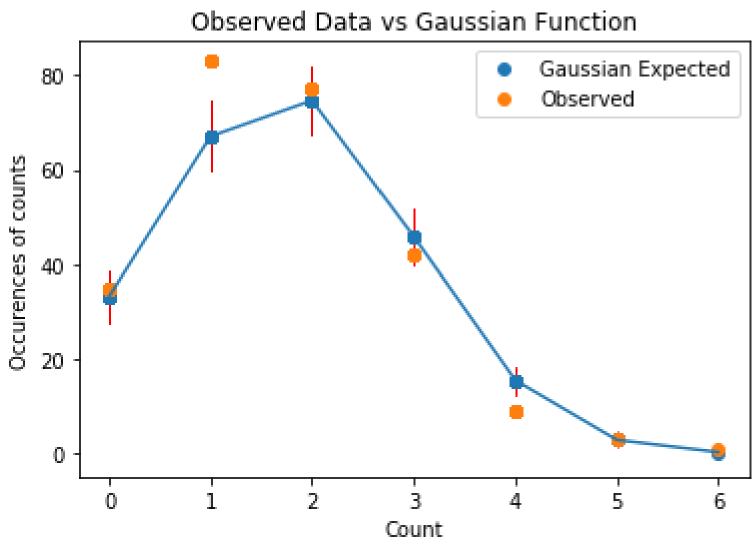
```
(0, 2.5)
```



In [171]:

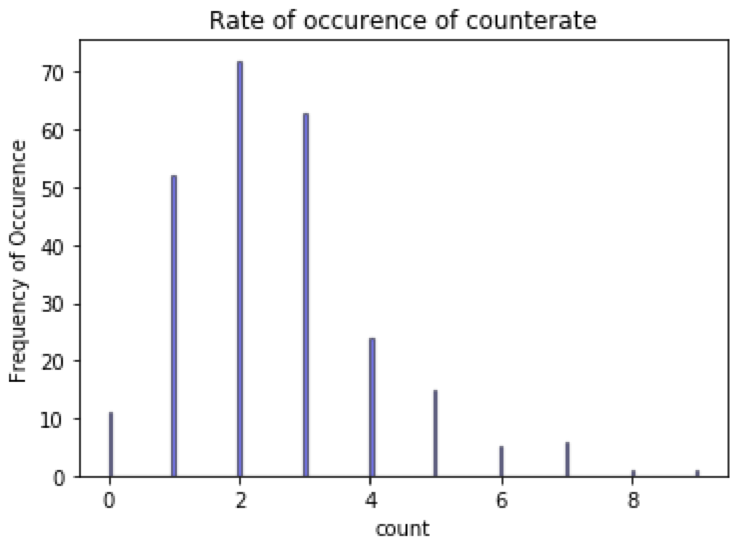
```
Gaussian(DataSet4)
```

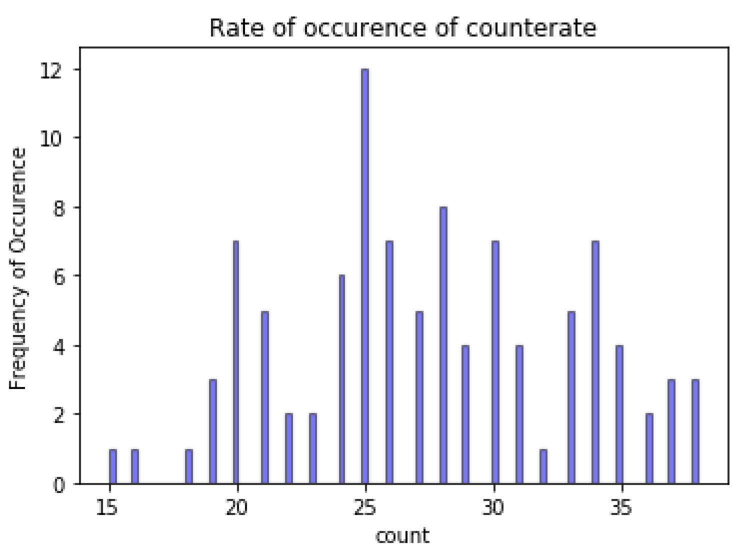
```
Total Number of Counts:  250
Mean:  1.68
Standard Deviation:  1.296148139681572
Variance:  1.6799999999999997
Uncertainty of the Mean:  0.07151824548848354
Uncertainty of the Variance:  0.11460116597551266
Degrees of Freedom: 5
Chi Square:  454.25788646258087
Ordinary Leasts Squares:  90.85157729251617
```

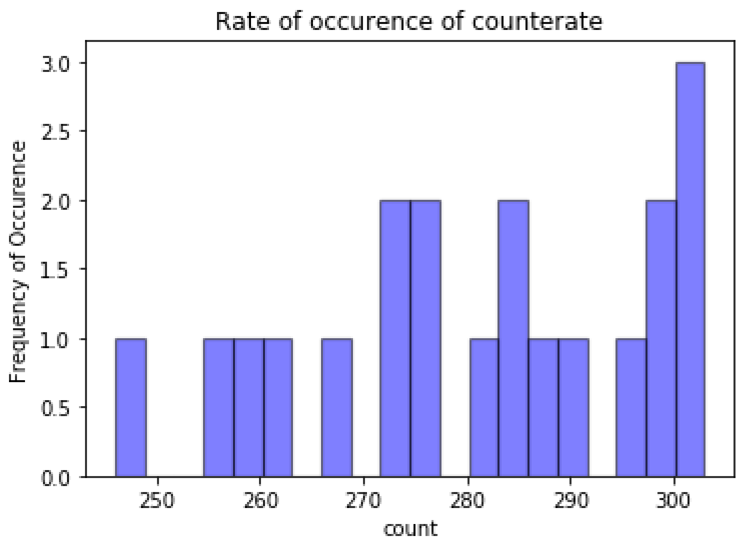


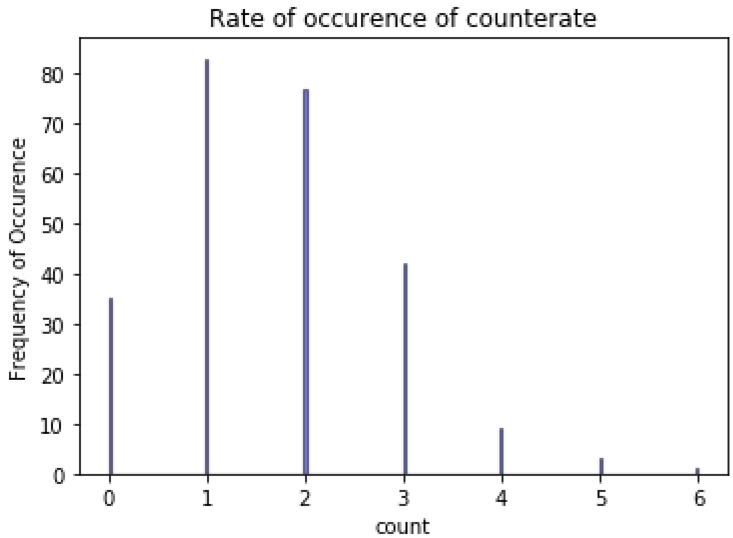
In [185]:

```
Histogram(DataSet1,Stat(DataSet1))
Histogram(DataSet2,Stat(DataSet2))
Histogram(DataSet3,Stat(DataSet3))
Histogram(DataSet4,Stat(DataSet4))
```






Rate of occurence of counterate
Frequency of Occurence
count
0.0
0.5
1.0
1.5
2.0
2.5
3.0
250
260
270
280
290
300


Rate of occurence of counterate
Frequency of Occurence
count
0
10
20
30
40
50
60
70
80
0
1
2
3
4
5
6

In [ ]: