PyLab 5 - Random Number Analysis with Radioactive Decay

Chris Compierchio

October 23, 2020

Abstract:

The purpose of this lab was to plot a histogram of the data detected from a decaying fiesta plate and plot a Poisson PMF and Gaussian distribution along with the data.

Introduction:

As mentioned in the abstract, a histogram was created using data from a decaying fiesta plate along with the corresponding Poisson and Gaussian distribution functions. This was done using the pyplot.hist(), poisson.pmf(), and scipy.stats.norm.pdf() functions in python.

Equipment:

- Fiesta Plate - The object being analyzed
- Geiger Counter - Measures the rate of gamma emissions

Procedure:

First, the Geiger counter and fiesta plate were set up so that the data could be recorded and plotted in real-time. This data was then imported into python where the histogram and distribution functions were plotted. Before plotting the data, however, the mean background radiation was subtracted from the data being analyzed in order to get a more accurate reading. This was done using the following line of code $y = y - mean$ where "mean" was calculated using the numpy.mean() function in python. The mu value and the standard deviation of the mu value were then found to plot the distribution functions on the histogram. Mu was taken to be the mean of the y values after the mean of the background data was subtracted. The standard deviation is the square root of mu, where the numpy.sqrt() function was used. The same process was then followed with the background data alone, which was compared to the plots of the y data (see Discussion secion).

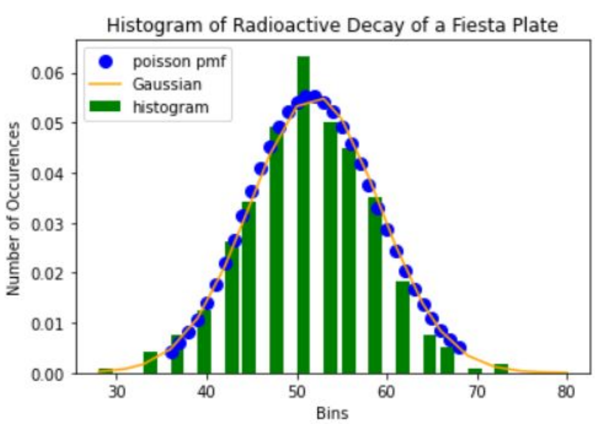
Results:

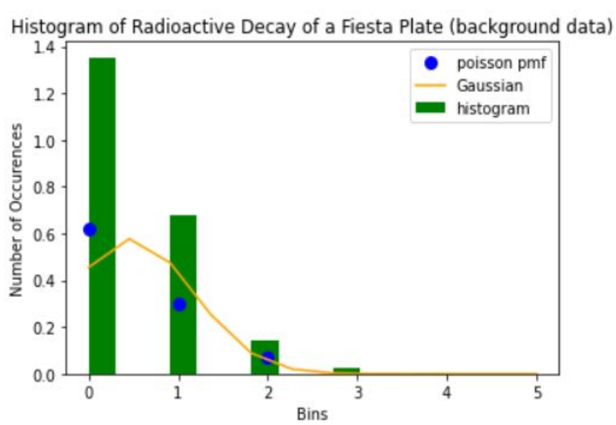
The count rates and errors were calculated using the following formula:

$$R = \frac{N}{\Delta t} \pm \frac{\sqrt{N}}{\Delta t}$$

Where N is the counts and Δt is the time interval of 3 seconds.

The histograms and distribution functions for the y data and background data were both plotted:





Discussion:

As seen in the plots above, the one for the y data (left) had a much better fit in terms of the distribution functions. This is because there were a sufficient amount of points plotted. This also causes the Gaussian and Poisson functions to overlap almost exactly, as they are expected to. The analysis of the background data, however, is much less fitting. This is because of the lack of data. Notice there is no center peak in the histogram for the background data like there is in the y data plot. This causes the distribution functions to only have one tail. This could perhaps be fixed if there was more data to analyze. This would also help the distribution functions fit the histogram better.

In [1]:

```
#import libraries
import numpy as np
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.stats import *
import scipy as sp
```

In [2]:

```
#import data
x, y = np.loadtxt("FiestaPlateActivity3sec20min.txt", skiprows = 2, delimiter = '\t', unpack = True
)
```

In [3]:

```
#import background data
x_bg, y_bg = np.loadtxt("FiestaBackground3sec20min.txt", skiprows = 2, delimiter = '\t', unpack = T
rue)
```

In [4]:

```
#calculate the mean background data
mean = np.mean(y_bg)
```

In [5]:

```
#subtract mean background data and convert to rates
y = (y - mean)

#convert x data to time
x = x*3

Rates = y/3
Rates_Error = np.sqrt(y)/3
```

In [6]:

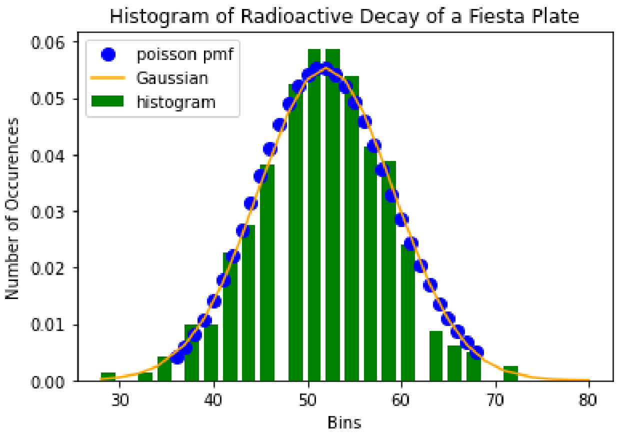
```
#calculate mu and sigma
mu = np.mean(y)
sigma = np.sqrt(mu)

#creat bins for first histogram
bins = np.round(np.linspace(28, 80, 25))
```

In [7]:

```
#plot histogram
plt.hist(y, bins,  histtype='bar', align='mid', orientation='vertical', width=1.5, color="green", d
ensity = True, label = "histogram")
plt.xlabel("Bins")
plt.ylabel("Number of Occurences")
plt.title("Histogram of Radioactive Decay of a Fiesta Plate")

#plot poisson function
z = np.arange(poisson.ppf(0.01, mu),

              poisson.ppf(0.99, mu))

plt.plot(z, poisson.pmf(z, mu), 'bo', ms=8, label='poisson pmf')

#plot gaussian distribution
plt.plot(bins, sp.stats.norm.pdf(bins, mu, sigma), color = "orange", label = 'Gaussian')
#show legend
plt.legend(loc = "upper left")
```

Out[7]:

```
<matplotlib.legend.Legend at 0x20a2709f670>
```



There are enough data points for the Gaussian and Poisson distributions to look approximately the same

In [8]:

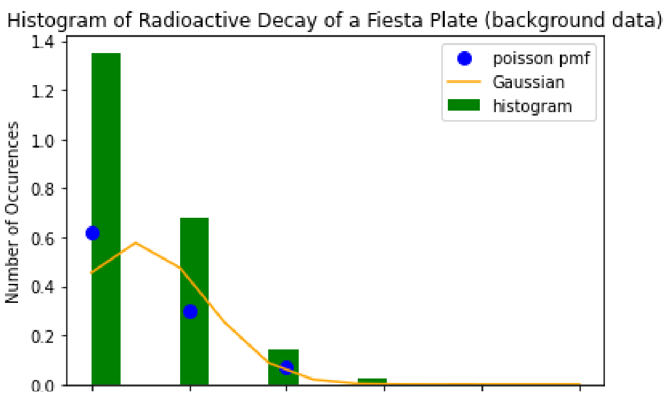
```
#define mu and sigma for background data
mu1 = np.mean(y_bg)
sigma1 = np.sqrt(mu1)

#create bins for background data
bins1 = np.linspace(0,5,12)
```

In [9]:

```
#plot hstogram for background data
plt.hist(y_bg, bins1,  histtype='bar', align='mid', orientation='vertical', width=0.3, color="green
", density = True, label = "histogram")
plt.xlabel("Bins")
plt.ylabel("Number of Occurences")
plt.title("Histogram of Radioactive Decay of a Fiesta Plate (background data)")

#plot poisson function
z1 = np.arange(poisson.ppf(0.01, mu1),

              poisson.ppf(0.99, mu1))

plt.plot(z1, poisson.pmf(z1, mu1), 'bo', ms=8, label='poisson pmf')

#plot cuassian distribution
plt.plot(bins1, sp.stats.norm.pdf(bins1, mu1, sigma1), color = "orange", label = "Gaussian")

#show legend
plt.legend(loc = "upper right")
```

Out[9]:

```
<matplotlib.legend.Legend at 0x20a270f6d00>
```



The distribution of the background data is less fitted because there are less data points. It also has the largest number of occurences at 0, meaning the normal distribution peak occurs at zero. The Fiesta Plate analysis has more data points over a wider range, therefore the normal distributions are more "fitting"