Random number analysis

Random events Random radioactive decays obey the rare event statistics given by the Poisson distribution (see Paragraph 6 from "Notes on Error Analysis" found at http://www.physics.utoronto.ca/~phy225h/web-pages/Notes_Error.pdf):

$$P_{n,\Delta t} = P_\mu(n) \tag{1}$$

where n stands for number of counts in any Δt and the Poisson *probability mass function* or *pmf* $P_\mu(n)$ is given by:

$$P_\mu(n) = e^{-\mu}\frac{\mu^n}{\Gamma(n+1)}. \tag{2}$$

where μ is the expected average number of counts per counting interval Δt and $\Gamma(n+1)$ is the Gamma function.

Note that this function is the **mass** function because the we're using discrete random variables (counts) and the *pmf* gives the probablity that a discrete random variable is exactly equal to a certain function. For continuous variables, we use the probablity density function (*pdf*). The integral of a *pdf* over a range of possible values (a,b) gives the probability that the continuous random variable falls within that range.

Background knowledge for exercise 5

Python : lists, arrays, numpy, SciPy, PyPlot, `curve_fit()`, `matplotlib`.
Error Analysis : Poisson Statistics.

1 Introduction

Lets look at an example: A radioactive source emits alpha particles at a rate of 3 particles per minute. Assuming a counting time of 2 minutes, the expected average number of counts will be the average rate of emission (3/minute) multipled by 3 minutes: $\mu = 6$. The probability of observing any number (n) of particles in the counting interval is given by:

$$P_6(n) = e^{-6}\frac{6^n}{n!} \tag{3}$$

In particular, the probability of observing exactly 4 particles in 2 minutes is given by:

$$P_6(4) = e^{-6}\frac{6^4}{4!} = 0.13 = 13\% \tag{4}$$

This means that the event happens once in about 8 trials.

2 Analysis of random data

Radioactive decay counts are commonly analyzed by displaying data in a bar-chart or histogram format, as seen in Figure 1. The bars correspond to count bins and the height of each bar is proportional to the count in the corresponding bin. A histogram module is available in numpy. Syntax can be found at: http://docs.scipy.org/doc/numpy/reference/generated/numpy.histogram.html. The `histogram` function doesn't plot the data, but allows you to change many configuration options in the histogram calculation. The corresponding `matplotlib.hist` (https://matplotlib.org/devdocs/api/_as_gen/matplotlib.pyplot.hist.html) function *does* plot the histogram, but doesn't provide access to as many configuration options.

Some input parameters for `histogram`:

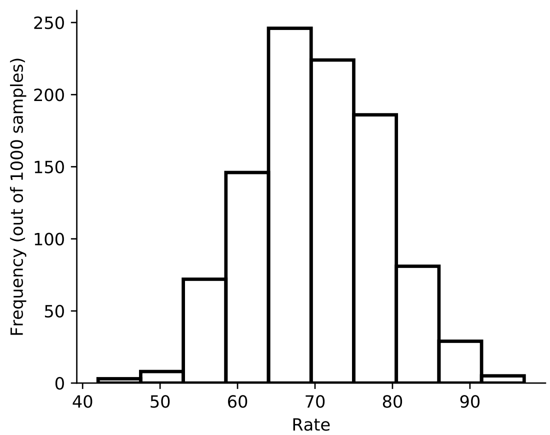

Figure 1: A typical histogram

a	: array-like input data
bins	: an integer defining the number of equal-width bins in the given range (10 by default),
range	: (float, float) is a float, meaning the lower and upper range of the bins,
normed	: a bool,optional. If True, the result is the value of the probability density function at the bin. Note: deprecated in Numpy 1.6, replaced by *density*

Outputs:

hist	: array: the values of the histogram
bin edges	: array of dtype float. For bins = 10 in the input, there will be 10 probability density values in n, 11 bin edges in bins.

3 The experiment

You will analyze the radioactive count from a Fiesta plate. Dinner plates from the Fiesta collection made before 1970 have a low amount of Uranium Oxide in the glaze. For more information, check: http://www.orau.org/ptp/collection/consumer%20products/fiesta.htm.

You will see the radioactive data acquisition in real time, but youll analyze a pre-acquired file.

The file: *Radioactive Activity-FiestaPlate3secDwell.txt* is located in the *2nd Yr Lab Files / Radioactive Decay* folder on the lab computers.

The file has two columns, *Sample Number* and *Number of Counts*. Time is obtained as: *Sample Number* $\times$ 3 seconds.

The background count can be found in another file in the same folder: *background20min.txt*. The sample time for the background data is 20 seconds, and occurred after the Fiesta plate data was taken. You should not compare the data in two files directly, but instead calculate the *average* background radiactivity before subtracting it from the sample activity.

3.1 Uncertainty in the Geiger counter

The Geiger counter follows *counting statistics* for determining the uncertainty. This is much like counting the number of heads in a series of coin-flip trials (the $\sqrt{N}$ rule). The rate is the number of events N divided by the sample time Δt. The best estimate of the count rate is given by,

$$R = \frac{N}{\Delta t} \pm \frac{\sqrt{N}}{\Delta t}\,. \qquad (5)$$

There is also background radiation to consider; you should have heard the counter beeping with no sample present. The counts from the two sources of radiation add together for our final reading,

$$N_{\mathrm{s}} = N_{\mathrm{T}} - N_{\mathrm{b}}\,. \tag{6}$$

The uncertainty can be calculated via standard error propagation, so

$$\begin{aligned}\sigma_{\mathrm{s}} &= \sqrt{\sigma_{\mathrm{T}}^2 + \sigma_{\mathrm{b}}^2}\\ &= \sqrt{N_{\mathrm{T}} + N_{\mathrm{b}}}\,.\end{aligned}$$

Finally, because the background radiation cannot be measured at the same time, we assume it has constant statistics, and use its mean in the calculations ($N_{\mathrm{b}} = \bar{N}_{\mathrm{b}}$). The mean background radiation must be subtracted from each data point before analysis.

4 The Python program

The program should be organized as follows:

- Import the required functions and modules. You can find the Poisson mass function in `scipy.stats.poisson`, and the Gaussian function in `scipy.stats.norm`. You should also import `matplotlib.pyplot`, `numpy` if needed.
- Load the data using `loadtxt()`.
- Analyze the Fiesta plate by writing a program to output the histogram.
- Plot the histogram of the count (or rate) data. You should modify the histogram's appearence, and normalize as needed.
- Add the Poisson probability mass function to qualitatively fit the data. The most appropriate value for μ can by taking the average value of all of your count data. This is the *Maximum Likelyhood Estimation* of the parameter.
- Add the Gaussian distribution to your plot. From the *Central Limit Theorem* the Poisson function should tend towards a Gaussian for large enough datasets (number of measurements). In this case the mean value of the Gaussian is μ, and the standard deviation $\sigma = \sqrt{\mu}$. Plot a Gaussian with the mean value and standard deviation from the measurements and compare it to the Poisson distribution. Are there enough data points for the Poisson and Gaussian functions look the same?

Submit all the plots, histogram with normpdf.

5 Bonus, 10 marks

Perform the same analysis on the background data alone. Plot the histogram of background count and plot the Gaussian and Poisson functions with appropriate values of μ and σ. Comment on any differences between this analysis and the analysis from the Fiesta plate.

Prepared by Ruxandra Serbanescu, 2016 Edited by Christopher Lee, 2017.