

```

# practicing fitting histograms
#As is true in many programming languages, a goal can be achieved
#by many methods
#The below example is just one of many possible ways to do exactly
#the same thing
import numpy as np
import LT.box as B
import LT_Fit.parameters as P # get the parameter module
import LT_Fit.gen_fit as G # load the genfit module
import scipy.special as ms #need this for the factorial function
                           #called in myfun1(x).

file_name = 'histoFitting.data'
f = B.get_file(file_name)

# get the data
# current
A = B.get_data(f, 'A')

# here histo is a method inside LT.box. There are other ways of
# making histograms as well

h2 = B.histo(A, (0.5, 10.5), 10)
# this means A is the data being
# histogrammed, (0.0, 10.0) is the range, and 10 is
# number of bins, obviously the bin width is 1.

h2.plot();
#histograms can be fitted with the build in function for the
#LT.box.histo object
#for example h2.fit()
# or h2.fit(2.0, 8.0) will fit the range between 2.0, and 8.0
#However, this is limited
#the example below provides a way to fit histograms with any user
# defined function, including
#linear, polynomial, gaussian, poisson, etc.....

#put the bin centers and bin contents to two arrays
hx = h2.bin_center
hy = h2.bin_content
dy = np.sqrt(hy)
print ("hy is:\n", hy)
print ("dy is:\n", dy)

```

```

mu = P.Parameter(2., 'mu')
norm = P.Parameter(10., 'norm')

#The function defined here is a pure poisson function. The intitial
# "guess" of the parameters
# are given above
def myfun1(x):
    value = norm()*mu()**x*np.exp(-mu())/ms.gamma(x+1.0)
    return value

#you need to provide the independent variable x, and the yvalue at x
#in the forms of arrays. in this case, it's the bin_center
# (hx, as defined above)
#being x, and bin_content (hy) being y
#the way the fitting is done is the same as if you are simply fitting
#a Y vs X plot

fit10 = G.genfit(myfun1, [mu, norm], x = hx, y = hy)

B.plot_line(fit10.xpl, fit10.ypl, color='red')
h2.plot()
B.pl.show()

```