Particle Project

November 27, 2019

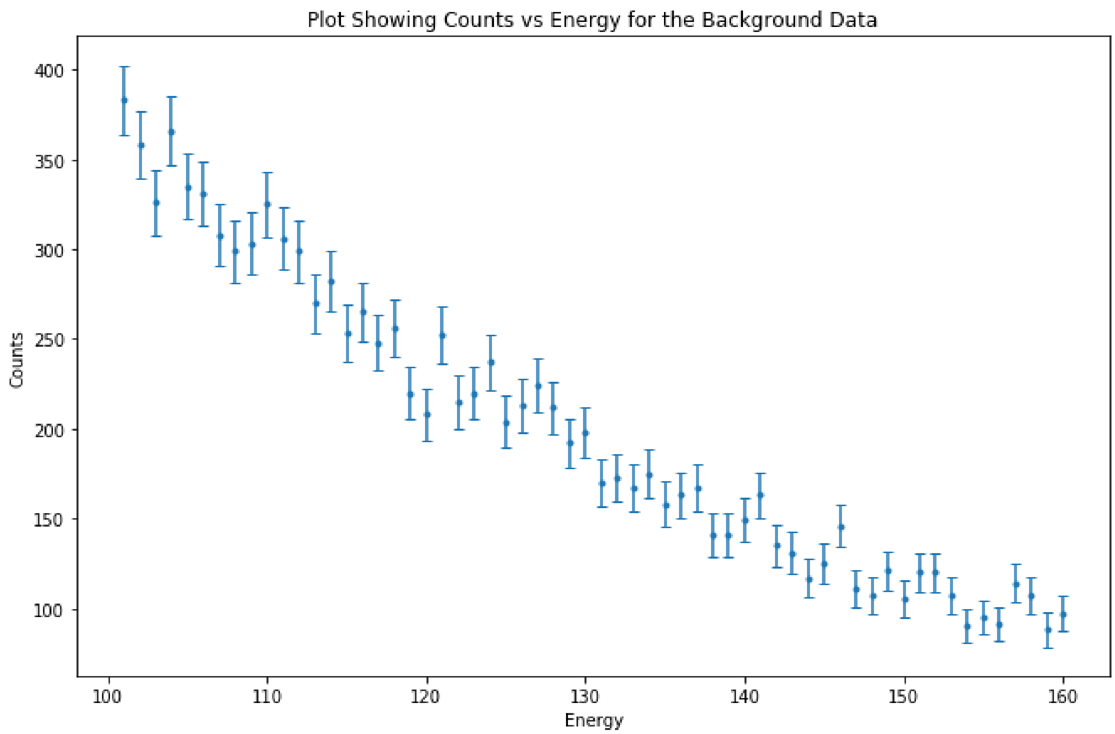
1 Question 1

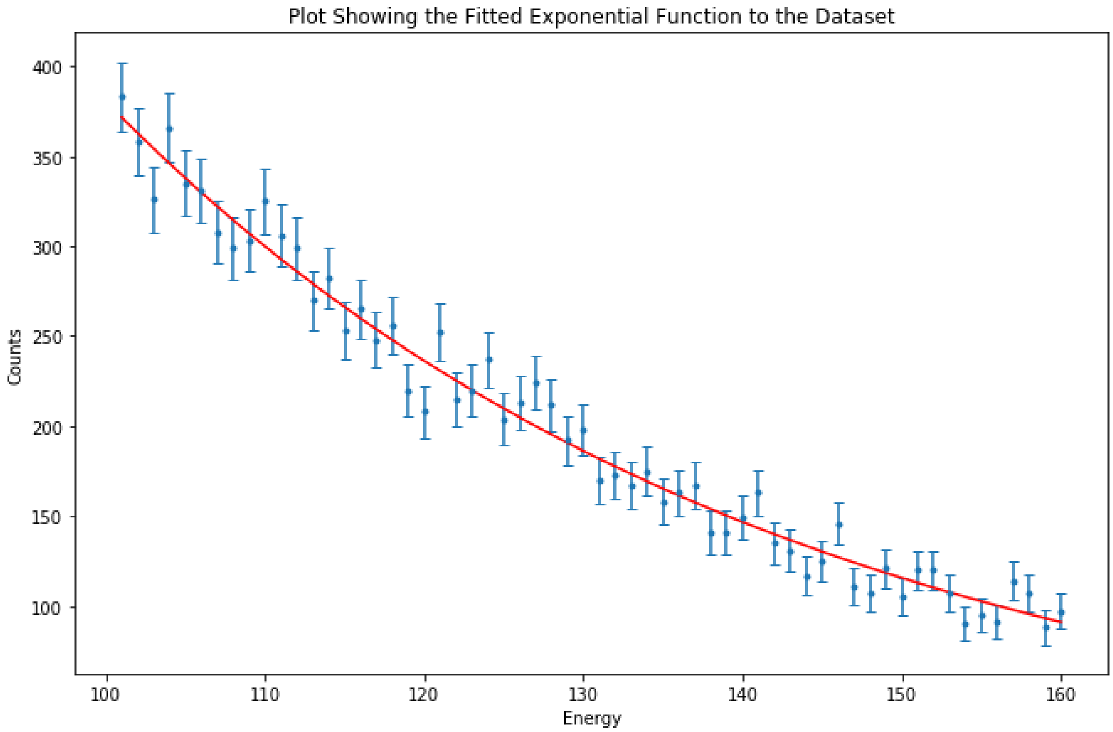
[6]:
```
from prettytable import PrettyTable
import scipy.optimize as optimize
%matplotlib inline
from scipy.io import loadmat
from scipy.stats import chisquare, chi2, norm
import matplotlib.pyplot as plt
import math
import numpy as np


## FUNCTIONS
def plotChiDist(chi,dof,):
    # Plot graphs.
    x = np.linspace(0,2.5*dof,100)

    # Pdf calcs
    CDF = chi2.pdf(x,dof)           # Curve
    y_chi2 = chi2.pdf(chi,dof)      # Sample Chi^2

    # Plot curve and points
    plt.plot(x,CDF, label="$\chi^2$ Distribution")
    plt.plot(chi,y_chi2,'r*', label="$\chi^2$" )
    plt.title("Plot Showing the $\chi^2$ Distribution for the fitted function")

    plt.xlabel('$\chi^2$')
    plt.ylabel("probability")
    plt.legend()
    plt.show()

def plotNormDist(m,s,chi):
    # Plot graphs.
    x = np.linspace(0,2.5*m,100)

    # Pdf calcs
    normDist = norm.pdf(x,m,s)           # Curve
```

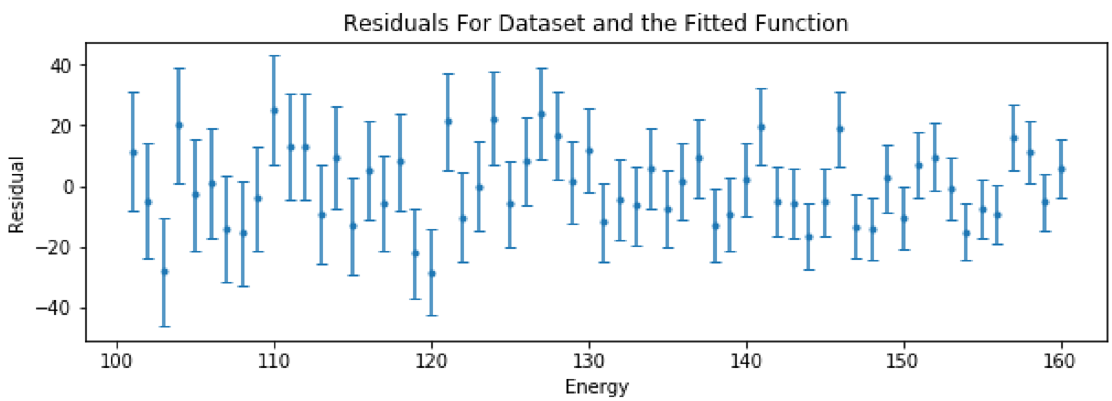
```
    y_chi2 = norm.pdf(chi,m,s)      # Sample Chi^2

    # Plot curve and points
    plt.plot(x,normDist, label="Normalised $\chi^2$ Distribution")
    plt.plot(chi,y_chi2,'r*', label="Normalised $\chi^2$" )
    plt.title("Plot Showing the Normalised $\chi^2$ Distribution for the fitted␣
 ↪function")

    plt.xlabel('$\chi^2$')
    plt.ylabel("probability")
    plt.legend()
    plt.show()

# Return upper and lower Confidence Interval
def calculateCI(CI,mean,sigma,doubleTail=True):
    if doubleTail: prob = (1-CI)/2
    else: prob = 1-CI
    return norm.isf(1-prob,mean,sigma),norm.isf(prob,mean,sigma)

def plotResidual(e,residual,error, fittedFunction = None):
    plt.figure(figsize=(10,3))
    plt.errorbar(e,residual,error, fmt=".", capsize=3)
    if fittedFunction is not None : plt.plot(e,fittedFunction, "r")
    plt.xlabel('Energy')
    plt.ylabel("Residual")
    plt.title("Residuals For Dataset and the Fitted Function")
    plt.show()

## LOAD DATASET
e=n=0
W = loadmat('ATLAS_DATA1.mat', mat_dtype=True, squeeze_me=True)
locals().update({k : W[k] for k in ['e', 'n']})

# PLOT DATASET
figure = plt.figure(figsize=(11,7))
n_error = np.sqrt(n)
plt.errorbar(e,n,n_error, fmt=".", capsize=3)
plt.xlabel('Energy')
plt.ylabel("Counts")
plt.title("Plot Showing Counts vs Energy for the Background Data")
plt.show()

# FIRST GUESSES
a = 1000
b = -0.01

def fitted_function(e,a,b):
```

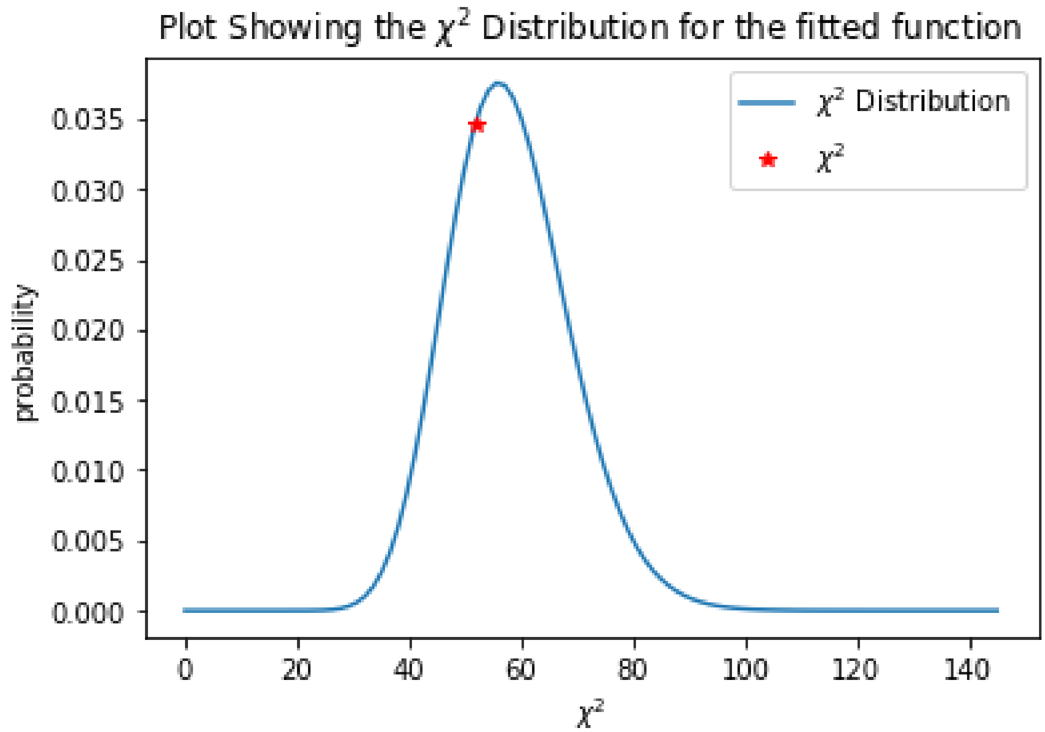
```python
    return a*np.exp(b*e)

params, params_covariance = optimize.curve_fit(fitted_function,e,n,p0=[a,b])
a_bkg = params[0]
b_bkg = params[1]
n_fitted = fitted_function(e,a_bkg,b_bkg)

# PLOT FITTED FUNCTION
figure = plt.figure(figsize=(11,7))
n_error = np.sqrt(n)
plt.errorbar(e,n,n_error, fmt=".", capsize=3)
plt.plot(e,n_fitted, "r")
plt.xlabel('Energy')
plt.ylabel("Counts")
plt.title("Plot Showing the Fitted Exponential Function to the Dataset")
plt.show()

residual = n-n_fitted

# PLOT RESIDUALS
plotResidual(e,residual,n_error)

## Chi^2 plot
dof = len(n)-2
chi = chisquare(n,n_fitted,ddof=dof)
prob = chi2.sf(chi[0],dof)
reduced_chi2 = chi[0]/(dof)
significance = calculateCI(prob,0,1)

plotChiDist(chi[0],dof)

#Normalised Chi^2
normalisedChi = np.sqrt(2*chi[0])
probNorm = norm.sf(normalisedChi,(2*dof-1)**0.5,1)
normalisedSig = calculateCI(probNorm,0,1)

plotNormDist((2*dof-1)**0.5,1,normalisedChi)

## Print Tables
table = PrettyTable(['Property','Value'])
table.add_row(["Chi^2","{0:1.4f}".format(chi[0])])
table.add_row(["Reduced Chi^2","{0:1.4f}".format(reduced_chi2)])
table.add_row(["P(x > chi^2)","{0:1.4f}".format(prob)])
table.add_row(["Significance","{0:1.4f} , {1:1.4f}".
 ↪format(significance[0],significance[1])])
print(table)
```

```
print("")
print("Normalised Values")
table = PrettyTable(['Property','Value'])
table.add_row(["Chi^2","{0:1.4f}".format(normalisedChi)])
table.add_row(["P(x > chi^2)","{0:1.4f}".format(probNorm)])
table.add_row(["Significance","{0:1.4f} , {1:1.4f}".
↪format(normalisedSig[0],normalisedSig[1])])
print(table)

## NUMBER OF EVENTS
n1 = sum(n)
```

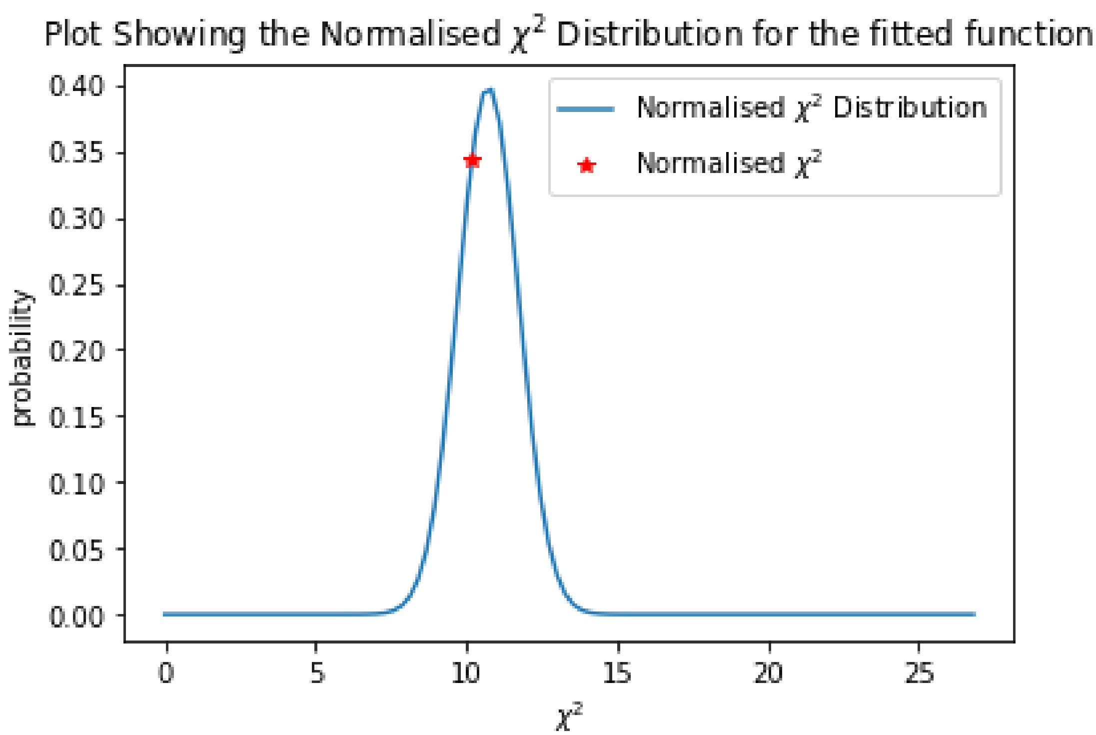



Plot Showing the Fitted Exponential Function to the Dataset
400
350
300
250
200
150
100
Counts
100
110
120
130
140
150
160
Energy


Residuals For Dataset and the Fitted Function
40
20
0
−20
−40
Residual
100
110
120
130
140
150
160
Energy



Plot Showing the χ^2 Distribution for the fitted function
χ^2 Distribution
χ^2
probability
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
20
40
60
80
100
120
140
χ^2



Plot Showing the Normalised χ^2 Distribution for the fitted function
Normalised χ^2 Distribution
Normalised χ^2
probability
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20
25
χ^2

Property	Value
Chi^2	51.7860
Reduced Chi^2	0.8929
P(x > chi^2)	0.7042
Significance	-1.0454 , 1.0454

Normalised Values

Property	Value
Chi^2	10.1770
P(x > chi^2)	0.7077
Significance	-1.0532 , 1.0532

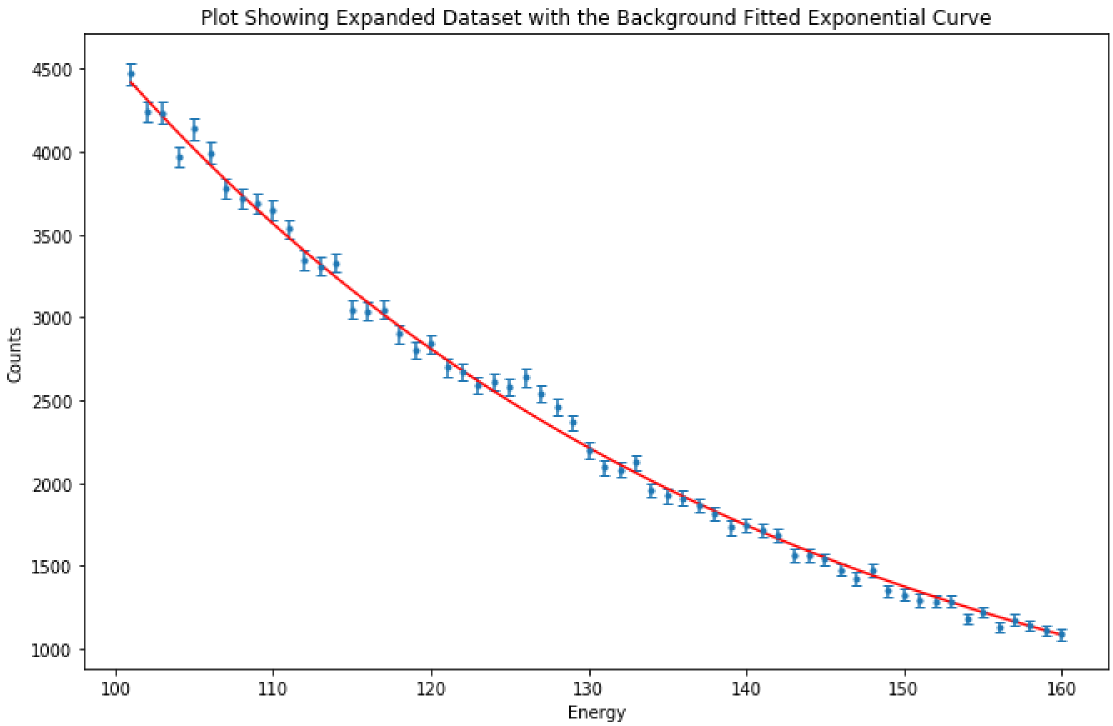
1.1 Conclusions

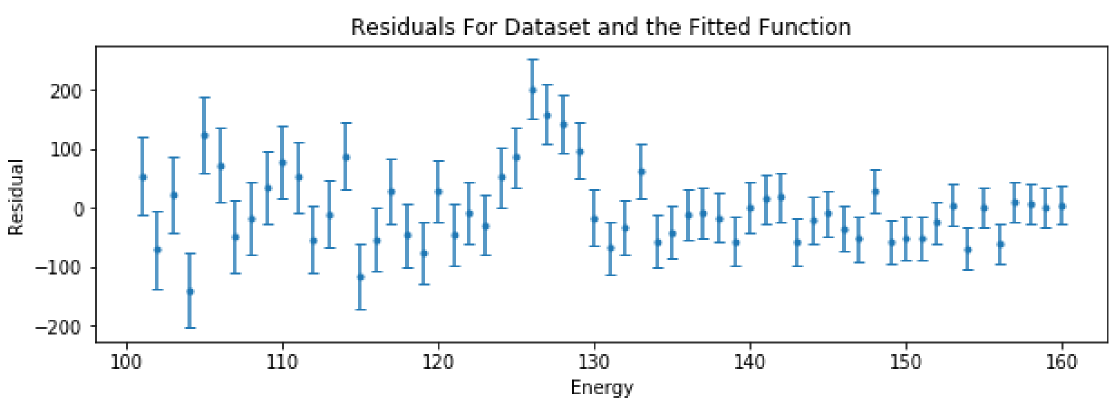
The background data can be fit with an exponential curve with a P value of 0.7077. Therefore providing a high level of confidence that this fit is acceptable for the data. The normalised χ^2 distribution provides a slightly higher significance value.

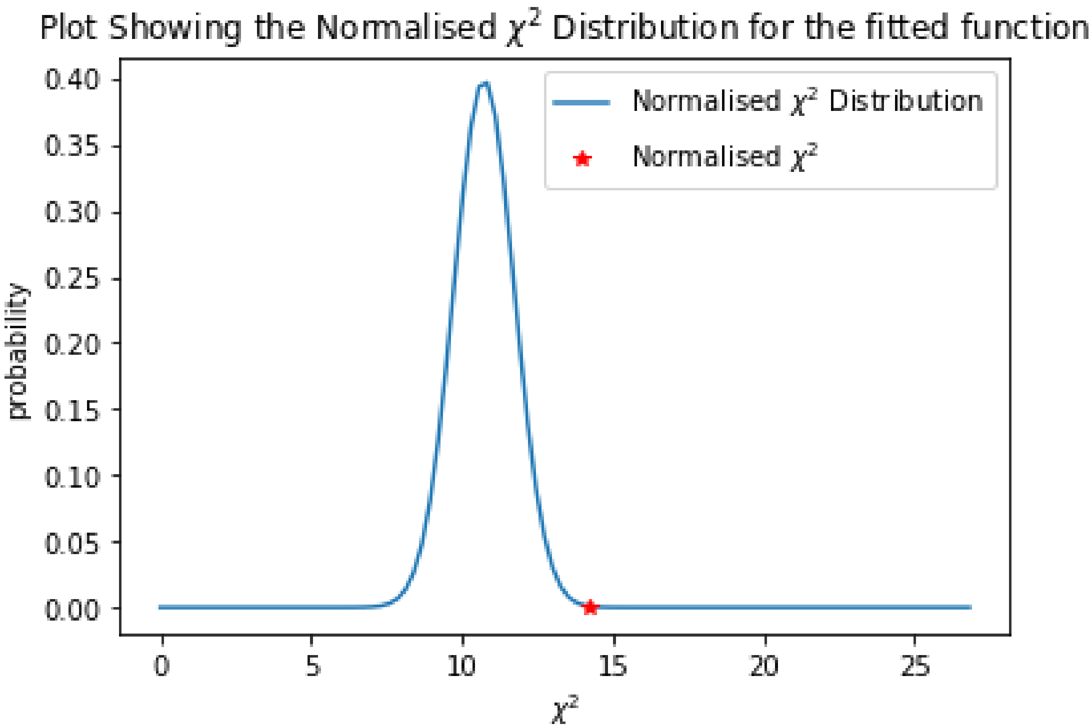
2 Question 2

[7]:
```
## LOAD DATASET
e=n=0
W = loadmat('ATLAS_DATA2.mat', mat_dtype=True, squeeze_me=True)
locals().update({k : W[k] for k in ['e', 'n']})

## Number of events in Dataset 2
n2 = sum(n)
a = a_bkg*(n2/n1)

# PLOT DATASET
figure = plt.figure(figsize=(11,7))
n_error = np.sqrt(n)
plt.errorbar(e,n,n_error, fmt=".", capsize=3)
n_fitted = fitted_function(e,a,b_bkg)
plt.plot(e,n_fitted, "r")
plt.xlabel('Energy')
plt.ylabel("Counts")
plt.title("Plot Showing Expanded Dataset with the Background Fitted Exponential␣
 ↪Curve")
plt.show()

residual = n-n_fitted

# PLOT RESIDUALS
plotResidual(e,residual,n_error)

## Chi^2 plot
dof = len(n)-2
chi = chisquare(n,n_fitted,ddof=dof)
prob = chi2.sf(chi[0],dof)
reduced_chi2 = chi[0]/(dof)
significance = calculateCI(prob,0,1)

#Normalised Chi^2
normalisedChi = np.sqrt(2*chi[0])
probNorm = norm.sf(normalisedChi,(2*dof-1)**0.5,1)
normalisedSig = calculateCI(probNorm,0,1)

plotNormDist((2*dof-1)**0.5,1,normalisedChi)

## Print Tables
print("Normalised Values")
table = PrettyTable(['Property','Value'])
table.add_row(["Chi^2","{0:1.4f}".format(normalisedChi)])
table.add_row(["P(x > chi^2)","{0:1.4f}".format(probNorm)])
```

```
table.add_row(["Significance","{0:1.4f} , {1:1.4f}".
    format(normalisedSig[0],normalisedSig[1])])
print(table)
```







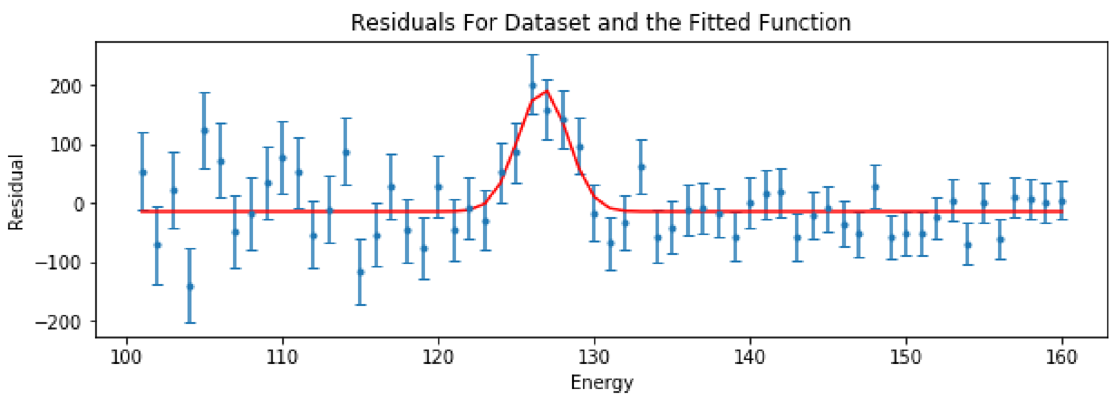

```
Normalised Values
+--------------+------------------+
|   Property   |      Value       |
+--------------+------------------+
|    Chi^2     |     14.2131      |
| P(x > chi^2) |      0.0002      |
| Significance | -0.0003 , 0.0003 |
+--------------+------------------+
```

2.1 Conclusions

It is clear from the P and Significance values that the dataset does not follow the background curve. It is way below the significance values of 5% and 1% and so is extremely significant.

3 Question 3

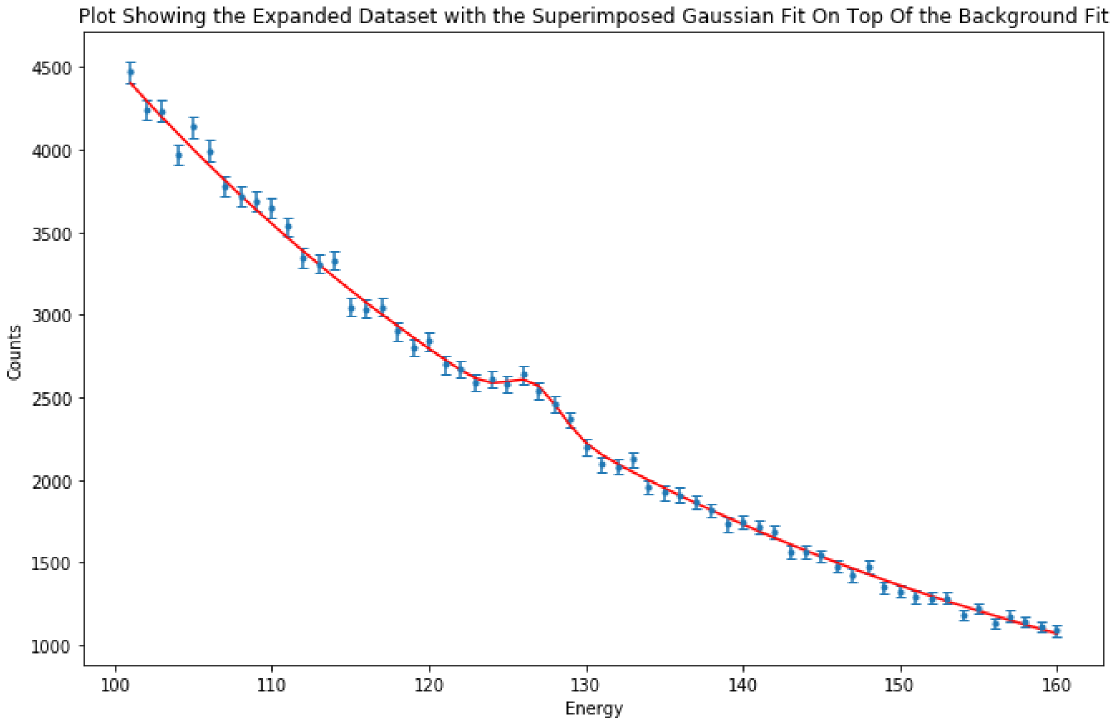
[8]:
```
# FIRST GUESSES

#Gaussian Terms
a2 = 200
m = 125
s = 2.5
c = 0

def fitted_function_gaussian(x,a,m,s,c):
    return a*np.exp(-(x-m)**2/(2*s**2))+c

params, params_covariance = optimize.
 ↪curve_fit(fitted_function_gaussian,e,residual,p0=[a,m,s,c])
fitted_gaussian =␣
 ↪fitted_function_gaussian(e,params[0],params[1],params[2],params[3])

n_fitted_peak = n_fitted+fitted_gaussian

# PLOT RESIDUALS
plotResidual(e,residual,n_error,fitted_gaussian)
```

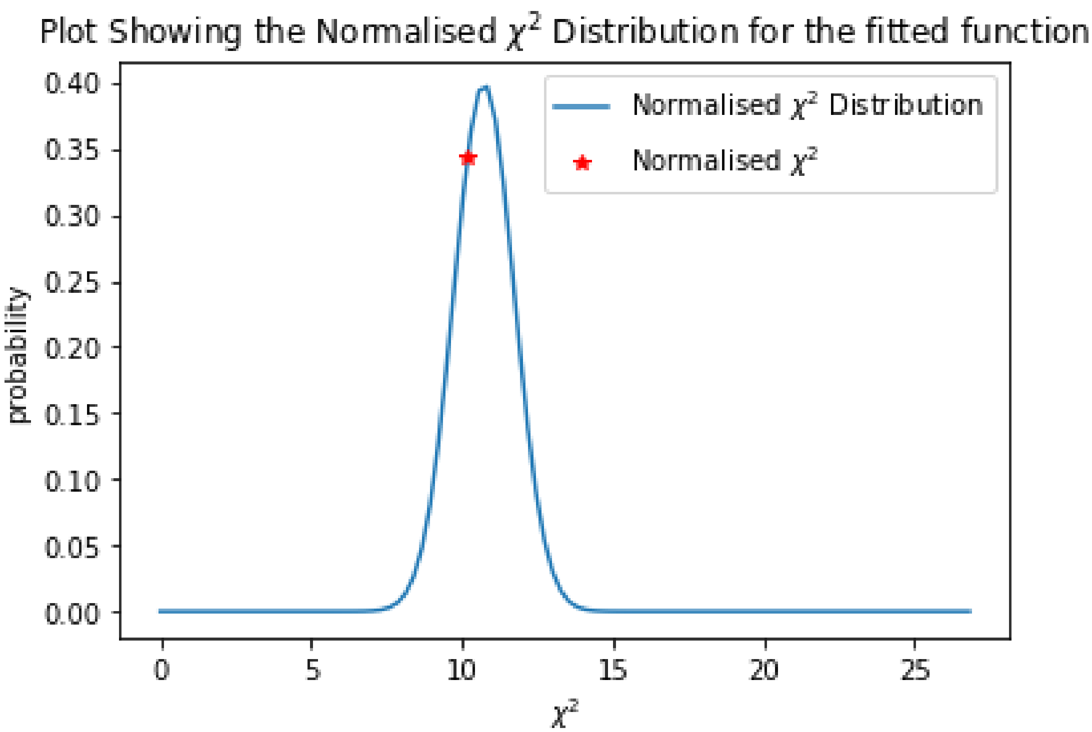


To calculate the Gaussian terms a curve was fitted to the residuals of the dataset, this could then be superimposed onto the exponential background curve. Note the *c* term, this gave the dataset a better fit. This could be due to the original exponential being an overestimate of the dataset and the *c* term is negative so corrects for that.

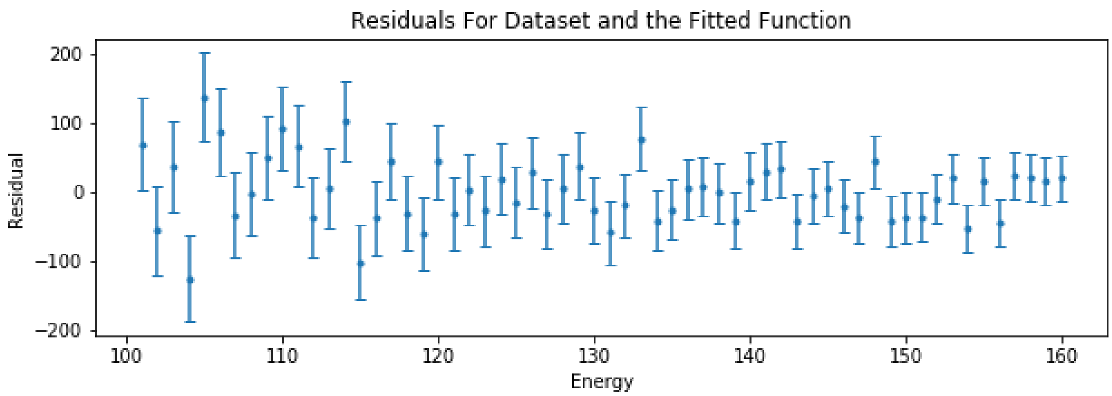
[9]:
```
# PLOT DATASET
figure = plt.figure(figsize=(11,7))
n_error = np.sqrt(n)
plt.errorbar(e,n,n_error, fmt=".", capsize=3)
plt.plot(e,n_fitted_peak, "r")
```

```
plt.xlabel('Energy')
plt.ylabel("Counts")
plt.title("Plot Showing the Expanded Dataset with the Superimposed Gaussian Fit␣
 ↪On Top Of the Background Fit")
plt.show()

## Chi^2 plot
dof = len(n)-2
chi = chisquare(n,n_fitted_peak,ddof=dof)
prob = chi2.sf(chi[0],dof)
reduced_chi2 = chi[0]/(dof)
significance = calculateCI(prob,0,1)

#Normalised Chi^2
normalisedChi = np.sqrt(2*chi[0])
probNorm = norm.sf(normalisedChi,(2*dof-1)**0.5,1)
normalisedSig = calculateCI(probNorm,0,1)

plotNormDist((2*dof-1)**0.5,1,normalisedChi)

## Print Tables
print("Normalised Values")
table = PrettyTable(['Property','Value'])
table.add_row(["Chi^2","{0:1.4f}".format(normalisedChi)])
table.add_row(["P(x > chi^2)","{0:1.4f}".format(probNorm)])
table.add_row(["Significance","{0:1.4f} , {1:1.4f}".
 ↪format(normalisedSig[0],normalisedSig[1])])
print(table)

residual = n-n_fitted_peak
plotResidual(e,residual,n_error)
```


Plot Showing the Expanded Dataset with the Superimposed Gaussian Fit On Top Of the Background Fit
Counts
Energy
4500
4000
3500
3000
2500
2000
1500
1000
100
110
120
130
140
150
160


Plot Showing the Normalised χ^2 Distribution for the fitted function
Normalised χ^2 Distribution
Normalised χ^2
probability
χ^2
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20
25

```
Normalised Values
+--------------+------------------+
|   Property   |      Value       |
+--------------+------------------+
|    Chi^2     |     10.1725      |
| P(x > chi^2) |      0.7093      |
| Significance | -1.0565 , 1.0565 |
+--------------+------------------+
```



[10]:
```python
## MASS OF HIGGS
mass = params[1]
N = params[0]
sig = params[2]
deltaMu = sig/np.sqrt(N)

print("Mass of Higgs:")
print("{0:1.2f} +/- {1:1.2f} GeV/c^2".format(mass,deltaMu))
```

```
Mass of Higgs:
126.72 +/- 0.11 GeV/c^2
```

3.1 Final Conclusions

Once the bump in the data is accounted for with a gaussian curve the significance of the fit is much better. From a P value of 0.0002 to 0.7093 signifies a much better fit that isn't rejected. The plot of the residuals afterwards now appears to be more uniform with no bell curve. The Higgs mass is as expected, but maybe a little to large to be consistent with the current mass of $125.18 \pm 0.16 Gev/c^2$