Data Science Analysis Assignment - 3

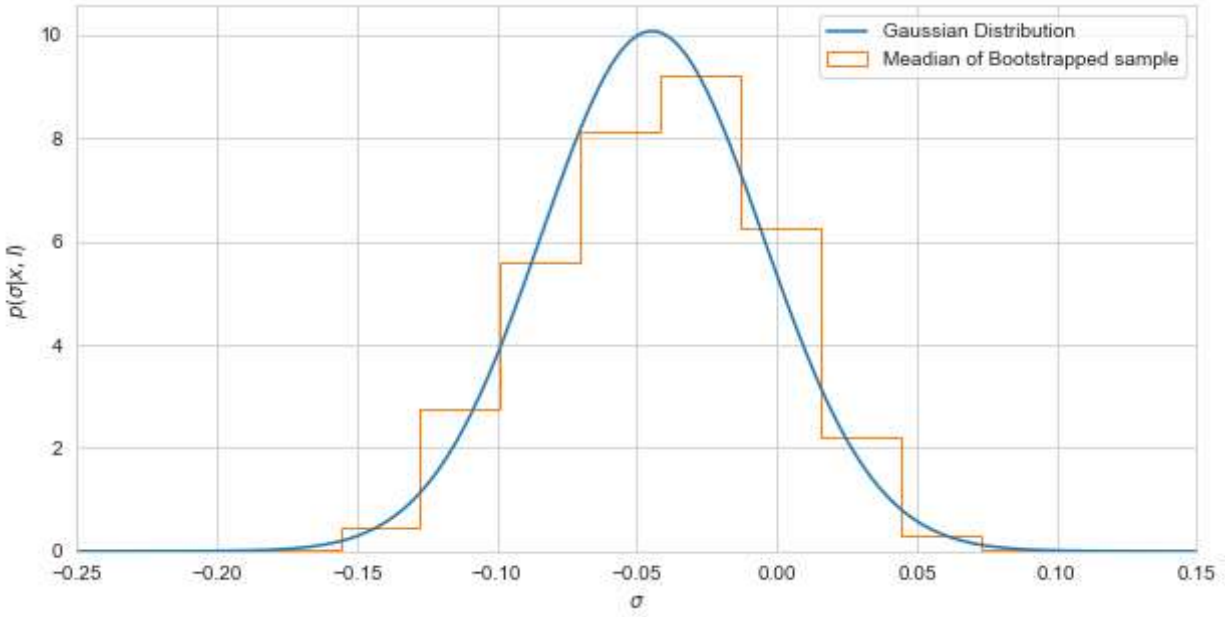
EE20BTECH11005 - Arumugam Swetha

In [1]:
```
#importing all the required libraries
import numpy as np
from scipy.stats import norm
from matplotlib import pyplot as plt
import seaborn as sb
from astroML.resample import bootstrap
from astroML.stats import median_sigmaG

import pandas as pd
from scipy.optimize import curve_fit
from scipy.stats import chi2
```

Q1

In [6]:
```
n = 1000  #no. of points
m = 10000 #no. of bootstraps

#sample values from a gaussian distribution
mu, sigma = 0, 1
np.random.seed(123)

data = norm(mu, sigma).rvs(1000)

#computing bootstrap resampling of the data
bs_data, _ = bootstrap(data, m, median_sigmaG, kwargs= {'axis' : 1})

#computing the theoretical distribution for the new data
bs_mean = np.mean(bs_data)
bs_std = np.sqrt(np.pi/(2*n))

x = np.linspace(-0.25, 0.15, 1000)
pdf = norm.pdf(x, loc = bs_mean, scale = bs_std)
```

```
#plotting the dists
fig, ax = plt.subplots(figsize=(10, 5))
ax.set_xlim(-0.25, 0.15)
sb.set_style('whitegrid')

plt.plot(x, pdf, label = "Gaussian Distribution")
plt.hist(bs_data, bins=10, histtype='step', density=True, label = "Meadian of Bootstrapped sample")

ax.set_xlabel(r'$\sigma$')
ax.set_ylabel(r'$p(\sigma|x,I)$')

plt.legend()
plt.show()
```



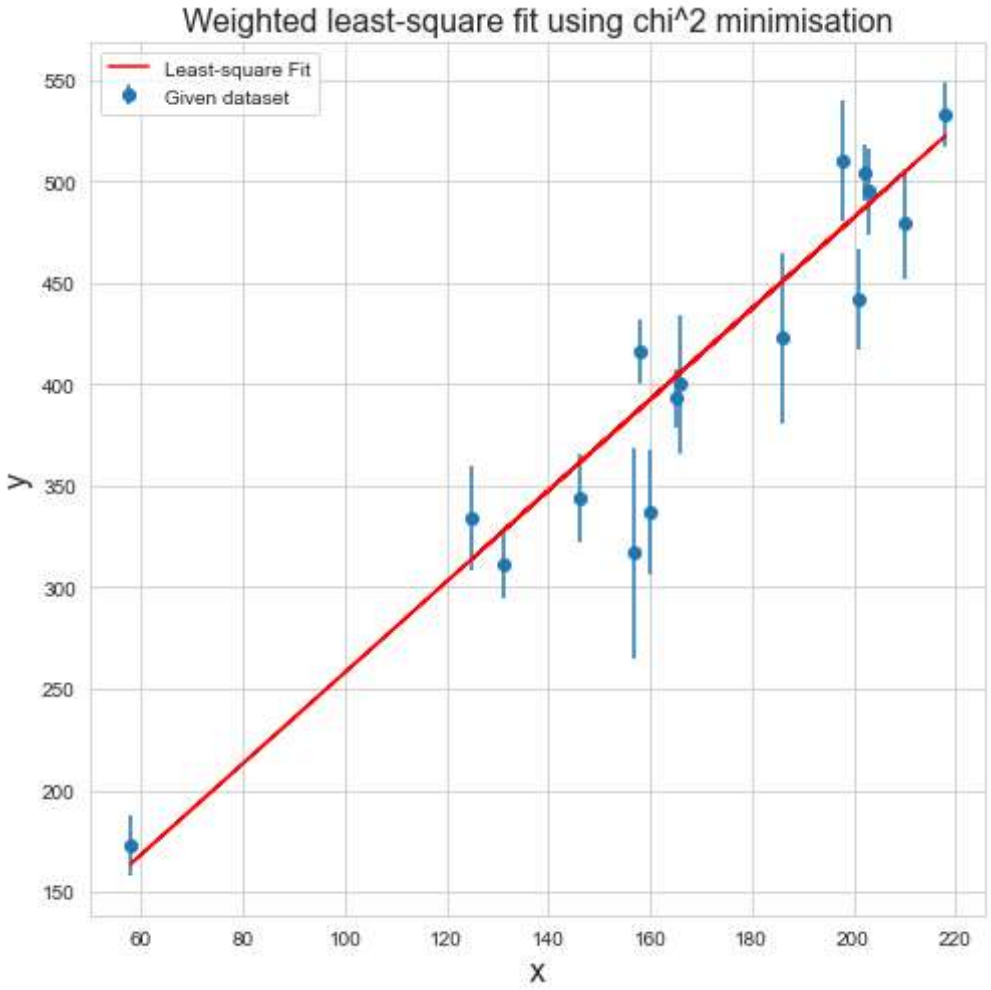
Q2

In [3]:

```
#importing data from csv file
df = pd.read_csv("D:\CLASSES\SEM 4\Data Science Analysis\A3\q2_data.csv", usecols = [0, 1, 2, 3])

x = df['x']
y = df['y']
y_err = df['error in y']
```

```
fig = plt.figure(figsize=(8,8))

plt.errorbar(x, y, yerr = y_err, fmt = 'o', label = 'Given dataset')

def fitting(x,m,b):
    return m*x + b

param, _ = curve_fit(fitting, x, y, sigma = y_err)
print("Fit done at m = %4.3f and b = %4.3f." %tuple(param))

plt.plot(x, fitting(x, *param), 'r-', label = 'Least-square Fit')

plt.title('Weighted least-square fit using chi^2 minimisation', fontsize=16)

plt.xlabel('x',fontsize=16)
plt.ylabel('y',fontsize=16)
plt.legend()

plt.show()
```

```
Fit done at m = 2.240 and b = 34.048.
```



Q3

In [7]:

```
models = ['correct errors','overestimated errors','underestimated errors','incorrect model']

N = 50
DOF = N - 1

chi2_dof = [0.96, 0.24, 3.84, 2.85]
chi2_values = np.multiply(chi2_dof,DOF)
```

```
p_values = []

for i in range(0,4):
    p_values.append(1 - chi2(DOF).cdf(chi2_values[i]))

print("p_values for the models are")

for i in range(0,4):
    print(" %s : %s" %(models[i],p_values[i]))
```

```
p_values for the models are
 correct errors : 0.5529264339960218
 overestimated errors : 0.9999999917009567
 underestimated errors : 0.0
 incorrect model : 1.2107292945984227e-10
```