Tanmay Garg

CS20BTECH11063

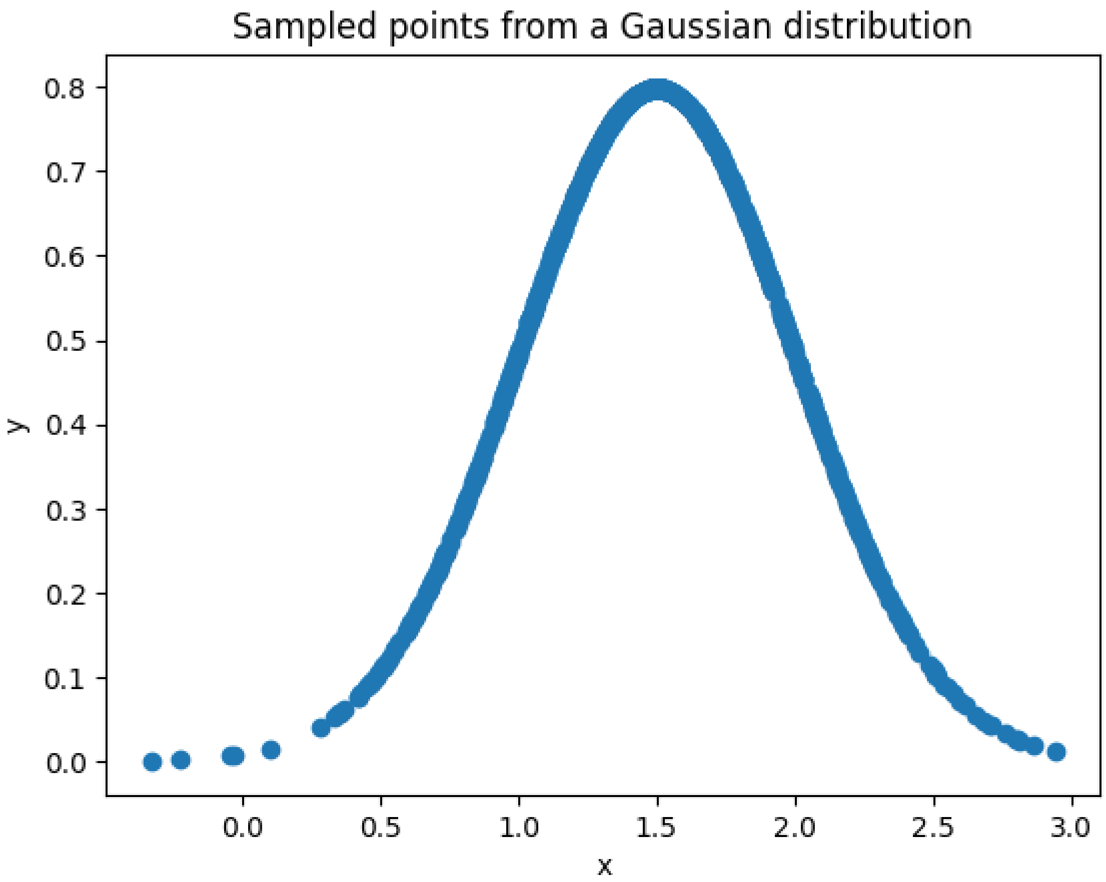
Data Science Analysis Assignment 1

Q1

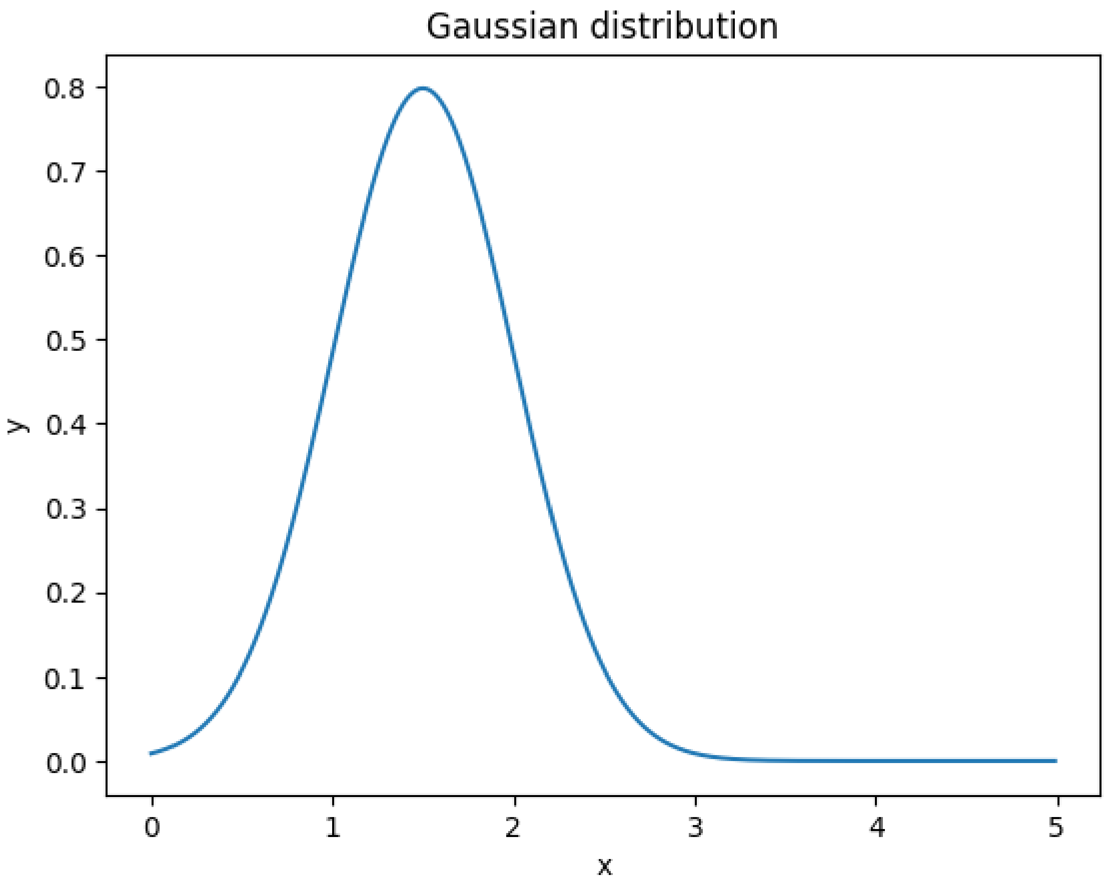
In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
import astroML
from astroML.stats import sigmaG

mu_gauss = 1.5
sigma_gauss = 0.5

# generate 1000 random numbers from a Gaussian distribution with mean of 1.5 and standard deviation of 0.5
# generated_samples = np.random.normal(mu_gauss, sigma_gauss, 1000)
normal_distribution = stats.norm(mu_gauss, sigma_gauss)
sampled_points = normal_distribution.rvs(1000)
generated_samples = sampled_points
y_sampled_points = normal_distribution.pdf(sampled_points)
plt.plot(sampled_points, y_sampled_points, 'o')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Sampled points from a Gaussian distribution')
plt.show()

# sample mean
sample_mean = np.mean(generated_samples)
print('sample mean: {}'.format(sample_mean))

# sample variance
sample_variance = np.var(generated_samples)
print('sample variance: {}'.format(sample_variance))

# sample skewness
sample_skewness = stats.skew(generated_samples)
```

```python
print('sample skewness: {}'.format(sample_skewness))

# sample kurtosis
sample_kurtosis = stats.kurtosis(generated_samples)
print('sample kurtosis: {}'.format(sample_kurtosis))

# sample standard deviation using MAD
sample_std_mad = 1.4826 * np.median(np.abs(generated_samples - np.median(generated_samples)))
print('sample standard deviation using MAD: {}'.format(sample_std_mad))

# sample standard deviation using sigma G
sample_std_sigma_g = sigmaG(generated_samples)
print('sample standard deviation using sigma G: {}'.format(sample_std_sigma_g))

# sample standard deviation using sigma G formula
q25 = np.percentile(generated_samples, 25)
q75 = np.percentile(generated_samples, 75)
sample_std_sigma_g_form = 0.7413*(q75 - q25)
print('sample standard deviation using sigma G formula: {}'.format(sample_std_sigma_g_form))

# Plot pdf of the Gaussian distribution
x = np.arange(0, 5, 0.01)
y = stats.norm.pdf(x, mu_gauss, sigma_gauss)
plt.plot(x, y)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Gaussian distribution')
plt.show()
```



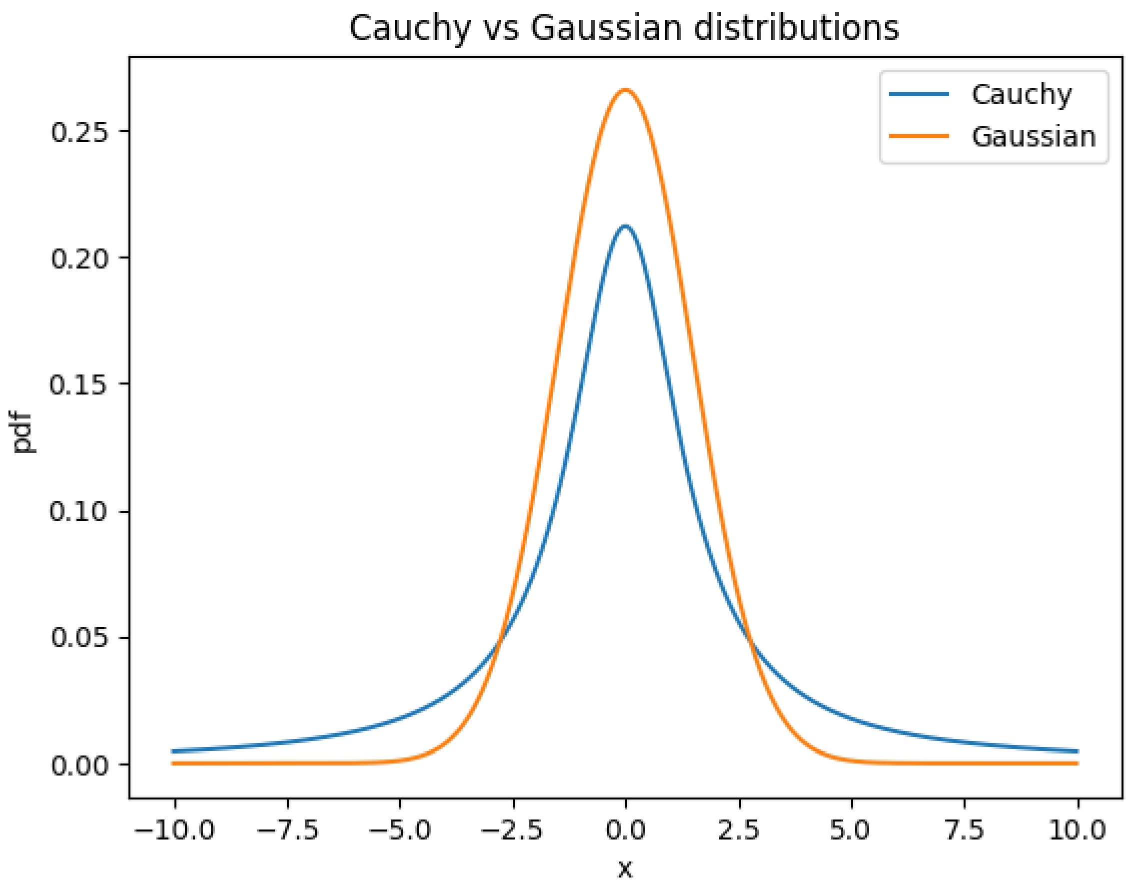

```
sample mean: 1.4927802294952257
sample variance: 0.24951610437768693
sample skewness: -0.05219274423104351
sample kurtosis: -0.06112995252829201
sample standard deviation using MAD: 0.5159749095007197
sample standard deviation using sigma G: 0.5201048225495241
sample standard deviation using sigma G formula: 0.5201040442858413
```



Q2

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats

mu_cauchy = 0
gamma_cauchy = 1.5
x = np.arange(-10, 10, 0.01)
y_cauchy = stats.cauchy.pdf(x, mu_cauchy, gamma_cauchy)

mu_gauss = 0
sigma_gauss = 1.5
y_gauss = stats.norm.pdf(x, mu_gauss, sigma_gauss)

# Plot the two distributions
```

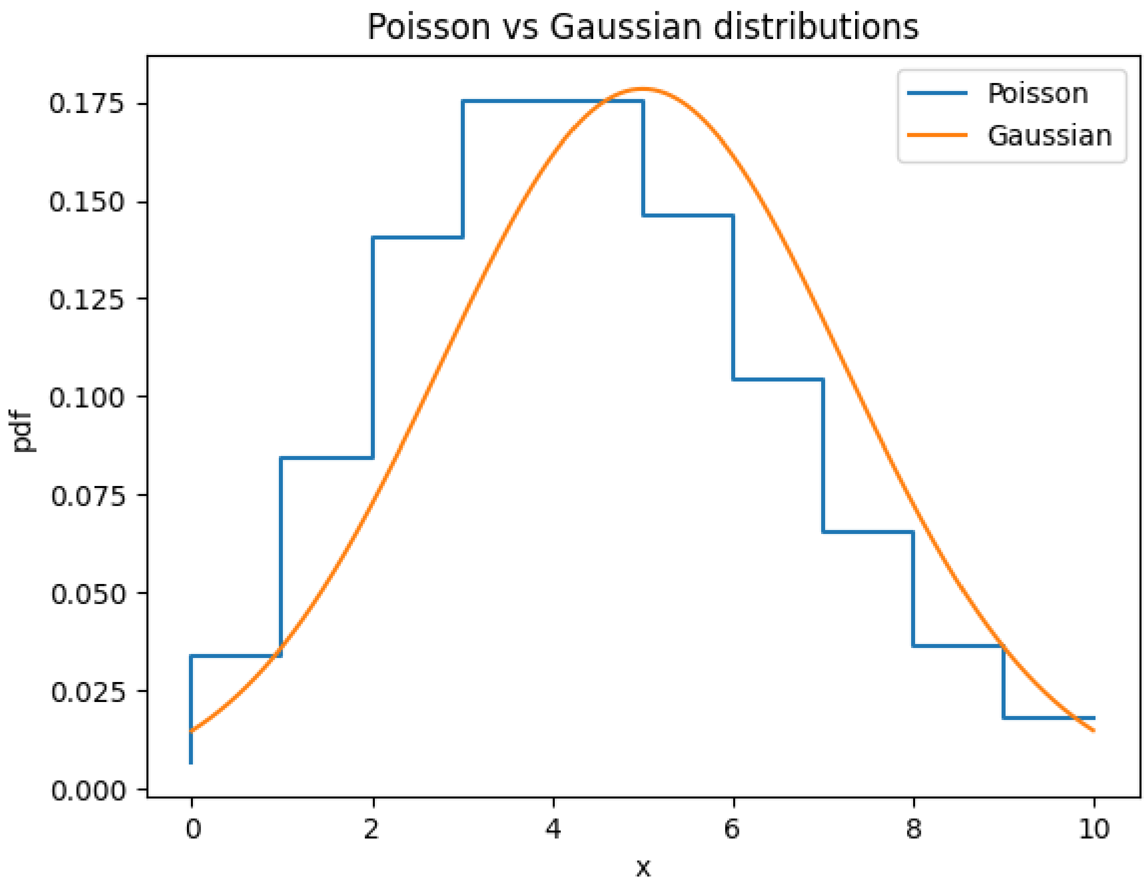
```python
plt.plot(x, y_cauchy, label='Cauchy')
plt.plot(x, y_gauss, label='Gaussian')
plt.legend()
plt.xlabel('x')
plt.ylabel('pdf')
plt.title('Cauchy vs Gaussian distributions')
plt.show()
```



Q3

In [3]:
```python
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats

x = np.arange(0, 10, 0.01)
```

```
mu_gauss = 5
sigma_gauss = np.sqrt(5)
y_gauss = stats.norm.pdf(x, mu_gauss, sigma_gauss)

mu_poisson = 5
y_poisson = stats.poisson.pmf([0,1,2,3,4,5,6,7,8,9,10], mu_poisson)

# Plot the two distributions
plt.step([0,1,2,3,4,5,6,7,8,9,10], y_poisson, label='Poisson')
plt.plot(x, y_gauss, label='Gaussian')
plt.legend()
plt.xlabel('x')
plt.ylabel('pdf')
plt.title('Poisson vs Gaussian distributions')
plt.show()
```



Q4

In [4]:
```
measure_val = [0.8920, 0.881, 0.8913, 0.9837, 0.8958]
measured_uncertainty = [0.00044, 0.009, 0.00032, 0.00048, 0.00045]
measure_val = np.array(measure_val)
measured_uncertainty = np.array(measured_uncertainty)

weighted_mean = np.average(measure_val, weights=1/(measured_uncertainty**2))
print('weighted mean: {}'.format(weighted_mean))
weighted_uncertainty = np.sqrt(1/np.sum(1/(measured_uncertainty**2)))
print('weighted uncertainty: {}'.format(weighted_uncertainty))
```

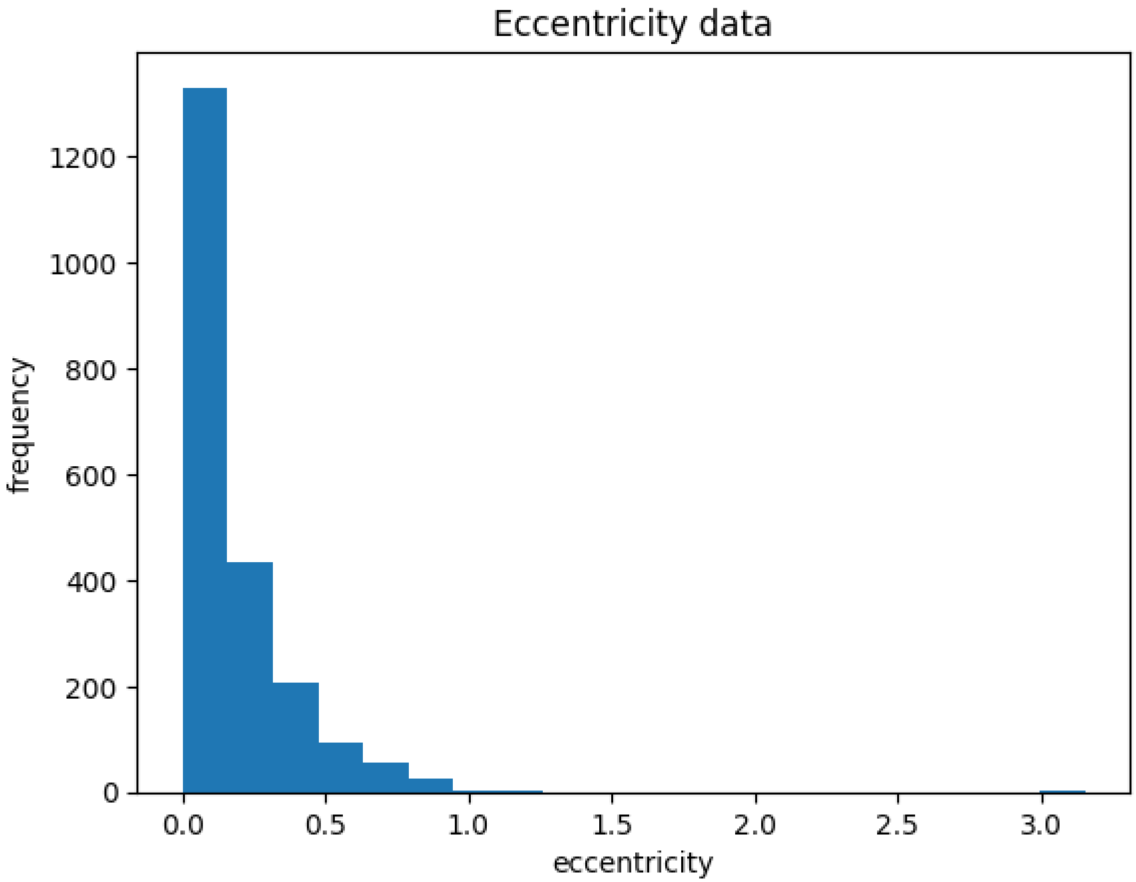
```
weighted mean: 0.9089185199574896
weighted uncertainty: 0.00020318737026848627
```

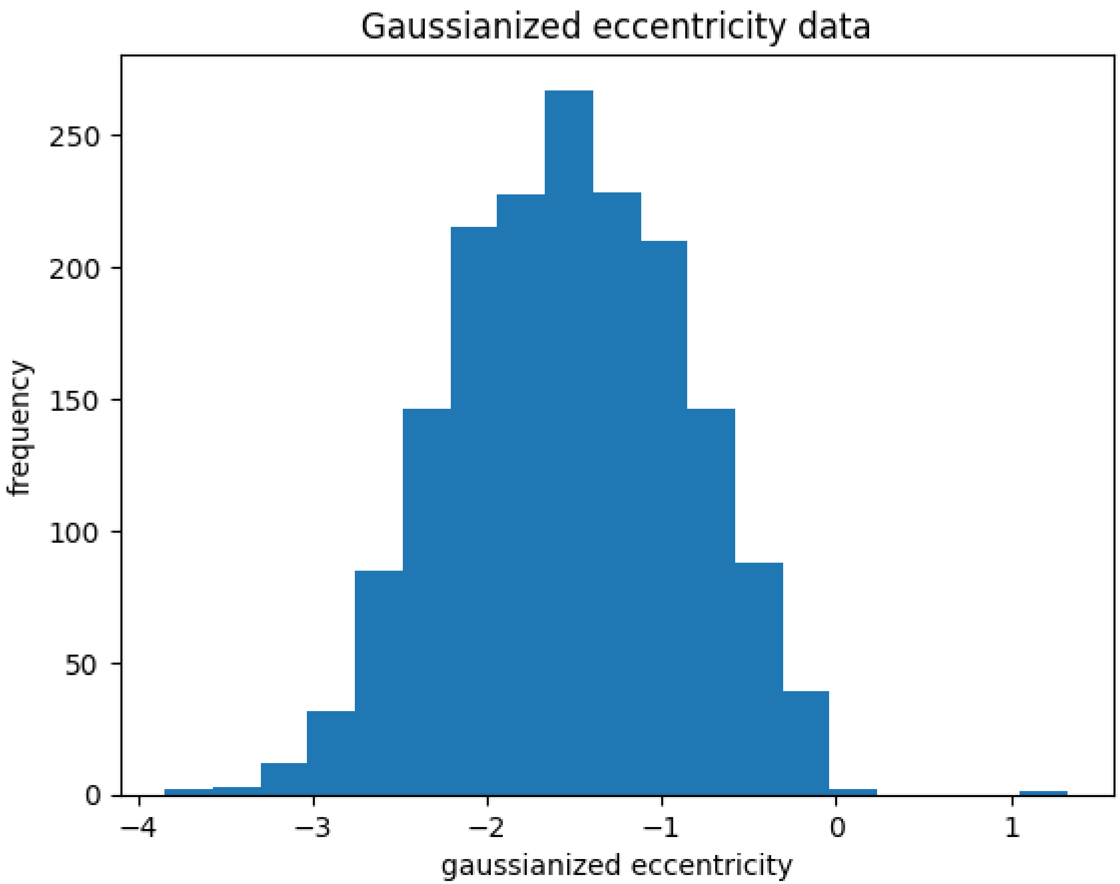
Q5

In [5]:
```
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
import pandas as pd

# read the data from the csv file
data = pd.read_csv('exoplanet.eu_catalog.csv')
tmp_dat = data
data = data['eccentricity'].dropna()

print(len(data))
# print only negative values of eccentricity
# print(data[data < 0])

# histogram of the eccentricity data
plt.hist(data, bins=20)
plt.xlabel('eccentricity')
plt.title('Eccentricity data')
plt.ylabel('frequency')
plt.show()

# gaussianize the eccentricity data
data_gaussianized = stats.boxcox(data[data>0])[0]

# histogram of the gaussianized eccentricity data
plt.hist(data_gaussianized, bins=19)
plt.xlabel('gaussianized eccentricity')
plt.ylabel('frequency')
```

```python
plt.title('Gaussianized eccentricity data')
plt.show()
```

2144




Gaussianized eccentricity data
frequency
gaussianized eccentricity
0
50
100
150
200
250
−4
−3
−2
−1
0
1