Data Science Analysis - Assignment 1

Varenya Upadhyaya

January 15, 2023

1. The following code plots the PDF (1) for a normal distribution with a mean of 1.5 and a standard deviation of 0.5

```
x=np.arange(0,3,0.01)
n = stats.norm.pdf(x,1.5,0.5)
plt.figure()
plt.plot(x,n,label='normal dist')
plt.title('Normal Distribution PDF')
plt.legend()
plt.savefig('1.png')
```

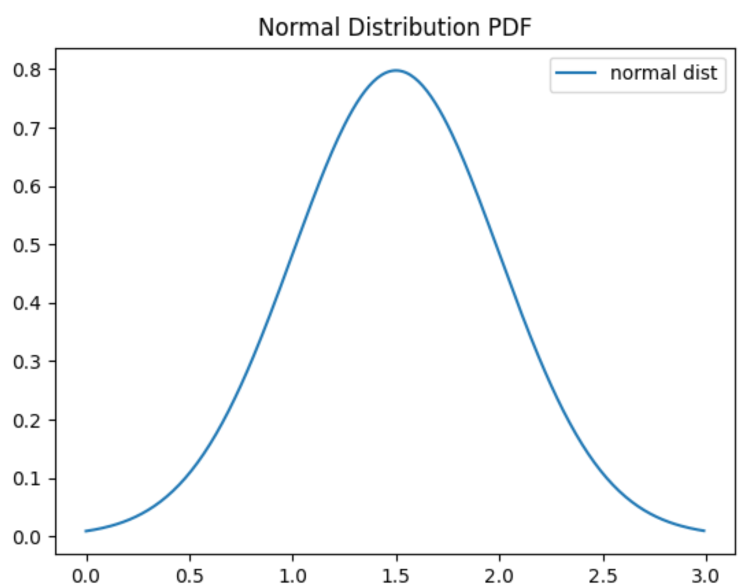

Figure 1: Normal Distribution

The mean, variance, skewness etc can be calculated using the following code:

```
dist = stats.norm(1.5,0.5)
sample = dist.rvs(1000)

mean = np.mean(sample)
variance = np.var(sample)
skewness = stats.skew(sample)
kurt = len(sample)*np.sum( (sample-np.mean(sample))**4 )/(np.sum((sample-np.mean(sample
    ))**2))**2
MAD = np.median(np.abs(sample-np.median(sample)))
std_dev_mad = 1.482 * MAD

sigma_g =  0.7413 * (np.percentile(sample,75)-np.percentile(sample,25))

print('mean =',mean)
```

```
print('variance =',variance)
print('skewness =',skewness)
print('kurtosis =',kurt)
print('Standard Deviation using MAD =',std_dev_mad)
print('Sigma_g =',sigma_g)
```

The output gives:

```
mean = 1.461047636900386
variance = 0.25171551000359926
skewness = 0.02043337907633425
kurtosis = 3.3334109056539014
Standard Deviation using MAD = 0.4920394817631893
Sigma_g = 0.48781923660020565
```

2. The following code plots Fig. 2

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

x=np.arange(-7,7,0.01)
dist = stats.norm(1.5,0.5)
dist_gaussian = stats.norm.pdf(x,0,1.5)
dist_cauchy = stats.cauchy.pdf(x,0,1.5)

plt.figure()
plt.plot(x,dist_gaussian,label='Gaussian')
plt.plot(x,dist_cauchy,ls='--',label='Cauchy')
plt.title('PDFs of Cauchy and Gaussian Distributions')
plt.legend()
plt.savefig('2.png')
```

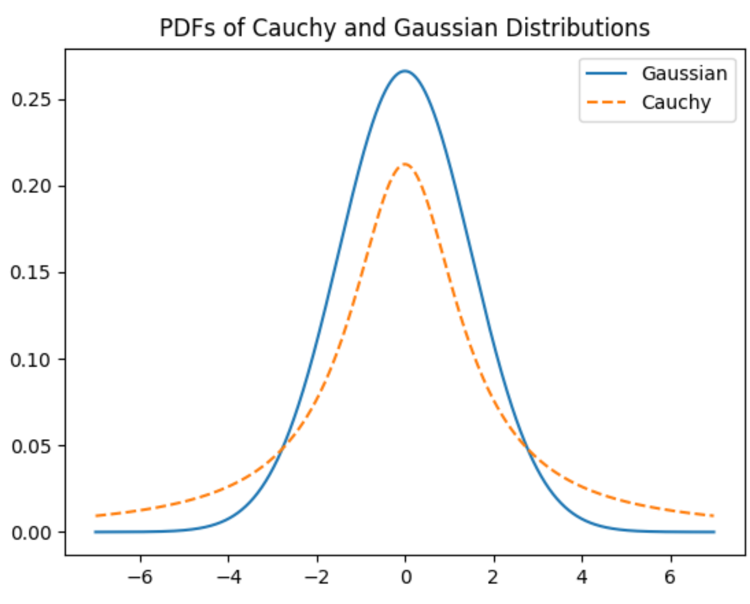

Figure 2: Cauchy v. Gaussian Distribution

3. The following code plots Fig. 3

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

```

x=np.arange(0,10,0.01)
dist = stats.norm(1.5,0.5)
dist_gaussian = stats.norm.pdf(x,5,np.sqrt(5))
dist_poisson = stats.poisson.pmf(x,5)

plt.figure()
plt.plot(x,dist_gaussian,label='Gaussian')
plt.plot(x,dist_poisson,label='Poisson')
plt.title('PDFs of Poisson and Gaussian Distributions')
plt.legend()
plt.savefig('3.png')
```

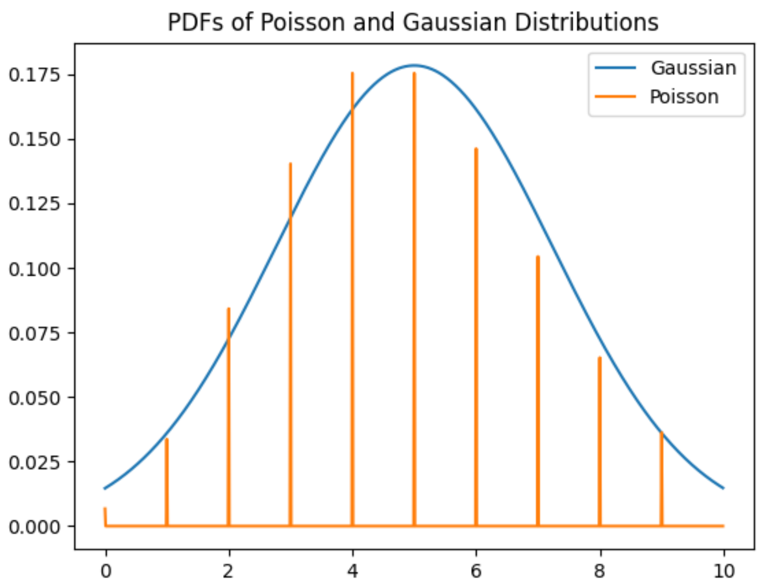

Figure 3: Poisson v. Gaussian Distribution

4. The values for the mean lifetime are given as:

$$x = 0.8920 \pm 0.00044; 0.881 \pm 0.009; 0.8913 \pm 0.00032; 0.9837 \pm 0.00048; 0.8958 \pm 0.00045 \tag{1}$$

The weighted mean can be calculated using the formula:

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i/\sigma_i^2}{\sum_{i=1}^{N} 1/\sigma_i^2} \tag{2}$$

$$\sigma_x^2 = \frac{1}{\sum_{i=1}^{N} 1/\sigma_i^2} \tag{3}$$

The following code computes the weighted mean and the corresponding uncertainty based on Eqs. (2) and (3).

```
import numpy as np

k = np.array([0.8920, 0.881, 0.8913, 0.9837, 0.8958])
e = np.array([0.00044,0.009,0.00032,0.00048,0.00045])

k_weighted = np.sum(k/e**2)/np.sum(1/e**2)
```

```
e_weighted = np.sqrt(1/np.sum(1/e**2))

print(k_weighted, e_weighted)
```

This gives (in $10^{-10}s$):

$$\bar{x} = 0.9089 \tag{4}$$

$$\sigma_x = 0.0002 \tag{5}$$

Weighted mean lifetime $= (0.9089 \pm 0.0002) \times 10^{-10}s$

5. The histograms for both the raw eccentricities and the gaussianized eccentricities can be seen in Fig. 4 The sample was gaussianized using the Box-Cox method and the following code illustrates the same:

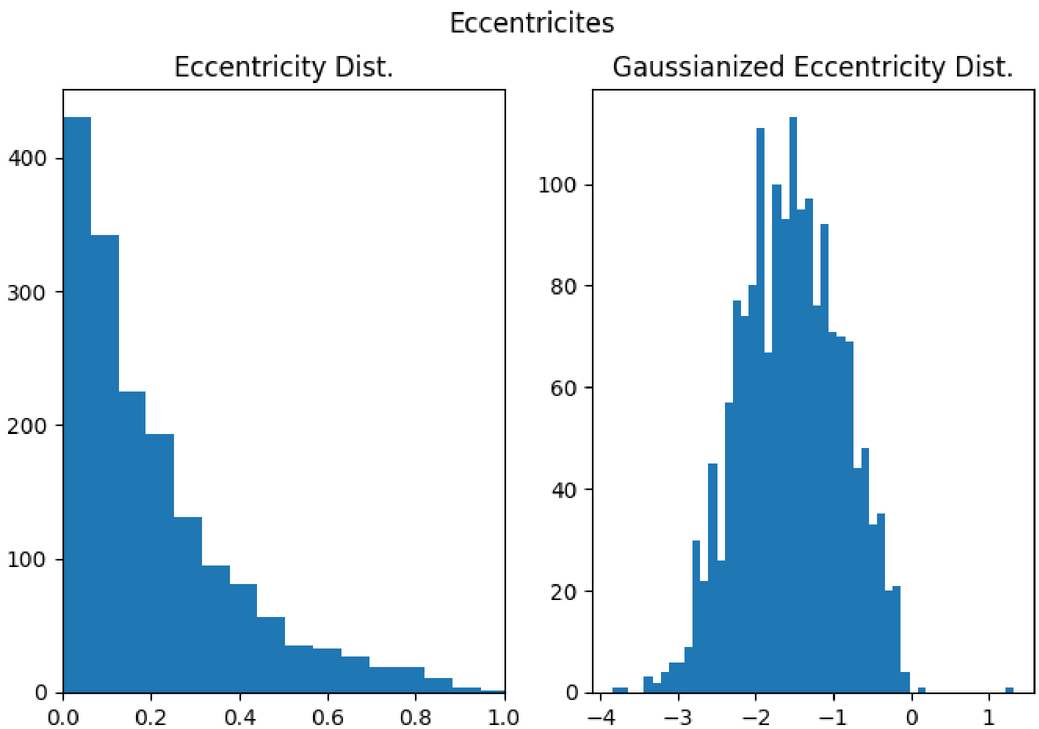

Figure 4: Eccentricities

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats

#reading the data
data = pd.read_csv("exoplanet.eu_catalog.csv")
e=data['eccentricity'].dropna().tolist()

#considering only the positive values (as is required for boxcox)
e = [x for x in e if x != 0]
e_norm,lamb = stats.boxcox(e)

print(e,e_norm) #printing out the samples

#plotting the histograms
fig, (ax1, ax2) = plt.subplots(1, 2)
fig.set_figheight(6)
fig.set_figwidth(8)
fig.suptitle('Eccentricites')
```

```
ax1.hist(e, 50, density=False)
ax1.set_title('Eccentricity Dist.')
ax1.set_xlim(0,1)
ax2.hist(e_norm, 50, density=False)
ax2.set_title('Gaussianized Eccentricity Dist.')
plt.savefig('5.png')
```

All the codes and figures used in this assignment can be found in this repository