Data Science Analysis Assignment - 1

Q1. Create 1000 draws from a normal distribution of mean of 1.5 and standard deviation of 0.5. Plot the pdf. Calculate the sample mean, variance,skewness, kurtosis as well as standard deviation using MAD and σG of these samples.

In [1]:

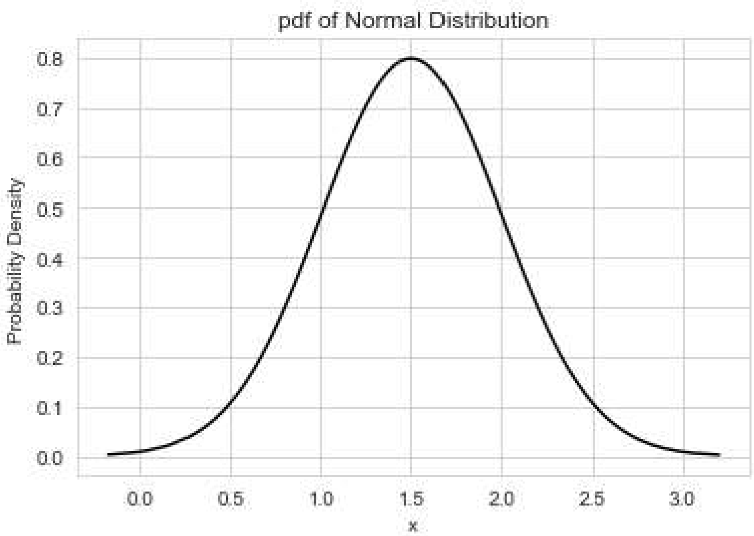
```
import warnings  ##to not show future warnings when the notebook is made into a pdf
warnings.filterwarnings('ignore')
```

In [11]:

```
#importing required libraries
from scipy.stats import norm, kurtosis, skew, median_abs_deviation
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb

#given mean and standard deviation
mu, sigma = 1.5, 0.5

#creating 1000 draws for normal dist.
data = np.random.default_rng().normal(mu, sigma, 1000)

#creating the pdf
pdf = norm.pdf(data , loc = 1.5 , scale = 0.5)

#Visualizing the distribution
sb.lineplot(data, pdf, color = 'black')
sb.set_style('whitegrid')

plt.xlabel('x')
plt.ylabel('Probability Density')
plt.title('pdf of Normal Distribution')
```

Out[11]:

```
Text(0.5, 1.0, 'pdf of Normal Distribution')
```



In [12]:

```
#calculating mvsk and printing them
mean = np.average(data)
var = np.std(data)**2
kur = kurtosis(data)
```

```python
sk = skew(data)

print("Sample mean = %0.5s \nVariance = %0.5s \nKurtosis = %0.6s \nSkewness = %0.6s"
      %(mean, var, kur, sk))
```

```
Sample mean = 1.502
Variance = 0.267
Kurtosis = -0.027
Skewness = -0.019
```

In [13]:

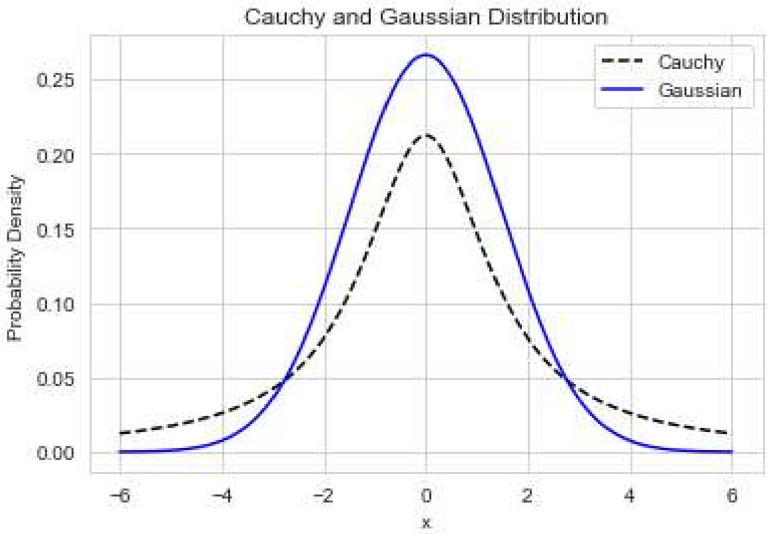
```python
#std using MAD
MAD = median_abs_deviation(data)
std_mad = MAD * 1.482

#importing library required for σG method
from astroML.stats import sigmaG

#std using sigmaG
std_sg = sigmaG(data)

print('Standard deviation using MAD and σG is %0.5s and %0.5s respectively.'
         %(std_mad, std_sg))
```

```
Standard deviation using MAD and σG is 0.520 and 0.521 respectively.
```

Q2. Plot a Cauchy distribution with μ=0 and γ=1.5 superposed on the top of a Gaussian distribution with μ=0 and σ=1.5. Use two different line styles to distinguish between the Gaussan and Cauchy distribution on the plot and also indicate these in the legends.

In [6]:

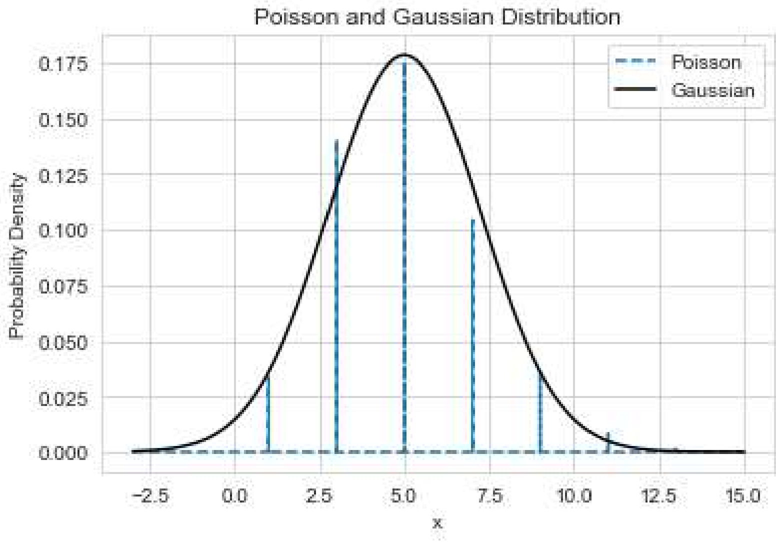
```python
#import library for Cachy dist.
from scipy.stats import cauchy

data1 = np.arange(-6, 6, 0.001)

#creating dist. for Cauchy
pdf1 = cauchy.pdf(data1, loc = 0, scale = 1.5)

#creating dist. for Gaussian
pdf2 = norm.pdf(data1, loc = 0, scale = 1.5)

#plotting the dist.
sb.set_style('whitegrid')

ax1 = sb.lineplot(data1, pdf1 , color = 'black')
ax1.lines[0].set_linestyle("--")
#using diff line styles to distinguish

ax2 = sb.lineplot(data1, pdf2 , color = 'blue')

#adding legend and title
plt.legend(['Cauchy','Gaussian'])
plt.title('Cauchy and Gaussian Distribution')

plt.xlabel('x')
plt.ylabel('Probability Density')

plt.show()
```



Q3. Plot Poisson distribution with mean of 5, superposed on top of a Gaussian distribution with mean of 5 and standard deviation of square root of 5. Use two different line styles for the two distributions and make sure the plot contains legends for both of them.

In [7]:

```
from scipy.stats import poisson

# creating the sample
data2 = np.linspace(-3, 15, 10000)

#creating the dist.
dist = poisson(5)
pmf3 = dist.pmf(data2)

pdf4 = norm.pdf(data2, loc = 5, scale = 5**0.5)

# plotting the sample
plt.plot(data2, pmf3,'--')
sb.lineplot(data2, pdf4 , color = 'black')

plt.xlabel('x')
plt.ylabel('Probability Density')

plt.legend(['Poisson','Gaussian'])
plt.title('Poisson and Gaussian Distribution')

plt.show()
```



Q4. The following were the measurements of mean lifetime of K meson (as of 1990) (in units of 10−10 s) : 0.8920±0.00044; 0.881±0.009; 0.8913±0.00032; 0.9837±0.00048; 0.8958± 0.00045. Calculate the weighted mean lifetime and uncertainty of the mean.

In [8]:

```
#making arrays for mean lifetime and errors
life_time = np.array([0.8920, 0.881, 0.8913, 0.9837, 0.8958])
error = np.array([0.00044, 0.009, 0.00032, 0.00048, 0.00045])

#calculating the wts
wts = 1./np.power(error,2.)

#calculating the weighted mean and uncertainity of the mean
wm = np.average(life_time, weights = wts)
em = (1/(sum(wts)))**0.5

print("Weighted mean lifetime is %0.6s and Uncertainity of the mean is %0.6s"
      %(wm,em))
```

```
Weighted mean lifetime is 0.9089 and Uncertainity of the mean is 0.0002
```

Q5. Download the eccentricity distribution of exoplanets from the exoplanet catalog http://exoplanet.eu/catalog/. Look for the column titled e, which denotes the eccentricity. Draw the histogram of this distribution. Then redraw the same histogram after Gaussianizing the distribution using Box-transformation either using scipy.stats.boxcox or from first principles using the equations shown in class or in arXiv:1508.00931. Note that exoplanets without eccentricity data can be ignored.

In [9]:

```
#importing required library
import pandas as pd

#reading the catalog and selecting the required column 'eccentricity'
ecc = pd.read_csv("D:\CLASSES\SEM 4\Data Science Analysis\downloads\ec.csv",
                  skipinitialspace=True, usecols = ['eccentricity'])

#making the plot space into parts for two diff graphs
fig, ax = plt.subplots(1, 2, figsize =(10, 4),tight_layout = True)
```

```
#plotting histogram for the given ecc
ax[0].hist(ecc)
ax[0].set_title("Histogram of eccentricities of exoplanets")

#removing rows with Nan and zero ecc values
ecc = ecc[ecc['eccentricity'].notna()]
ecc = ecc[(ecc[['eccentricity']] != 0).all(axis=1)]

##Gaussianizing the data##
#importing required library
from scipy import stats

#using box-cox method to gaussianize the eccentricity distribution
fitted_data, fitted_lambda = stats.boxcox(ecc['eccentricity'])

#plotting histrogram after gaussianizing ecc values
ax[1].hist(fitted_data, color = 'green')
ax[1].set_title("Histogram of gaussianised eccentricities of exoplanets")

plt.show()
```