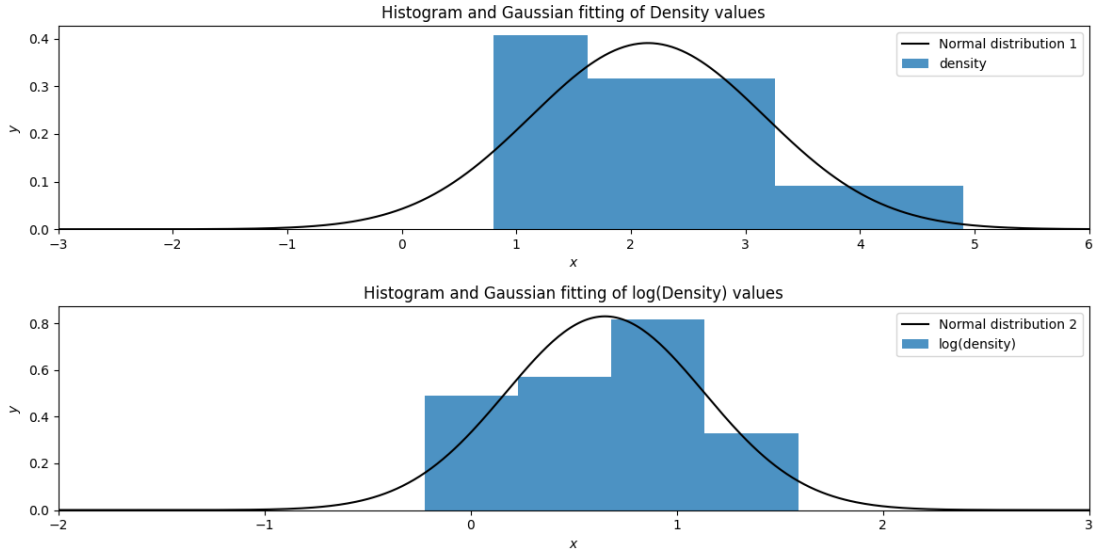
```
#QUESTION-1

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


data = pd.read_excel("C:\\Users\\Heera Baiju\\Downloads\\Data
Science\\Assignments\\Assignment 5\\Question 1 data.xlsx")
x = data['Dens']


a = stats.shapiro(x)
b = stats.shapiro(np.log(x))

print('Shapiro-wilk test of density, P values = :', a[1])
print('Shapiro-wilk test of log(density), P values = :', b[1])

mn1, sd1 = stats.norm.fit(x)
mn2, sd2 = stats.norm.fit(np.log(x))
t=np.linspace(-7, 7, 1000)
normal_distribution_1 = stats.norm.pdf(t, mn1, sd1)
normal_distribution_2 = stats.norm.pdf(t, mn2, sd2)

fig, ax = plt.subplots(2, 1, figsize = (6,6))
plt.xlim(-1.5, 3.5)
ax[0].plot(t, normal_distribution_1 , 'k-', label = 'Normal distribution 1')
ax[0].hist(x, density = True, histtype = 'stepfilled', alpha = 0.8,
label='density', bins = 'fd')
ax[0].legend(loc = 'upper right')
ax[0].set_xlim([-3, 6])
ax[0].set_title('Histogram and Gaussian fitting of Density values')
ax[0].set_xlabel('$x$')
ax[0].set_ylabel('$y$')
ax[1].plot(t, normal_distribution_2, 'k-', label='Normal distribution 2')
ax[1].hist(np.log(x), density = True, histtype='stepfilled', alpha = 0.8,
label='log(density)', bins='fd')
ax[1].legend(loc = 'upper right')
ax[1].set_xlim([-2, 3])
ax[1].set_title('Histogram and Gaussian fitting of log(Density) values')
ax[1].set_xlabel('$x$')
ax[1].set_ylabel('$y$')
plt.tight_layout()
plt.show()
```

OUTPUT



Shapiro-wilk test of density, P values = : 0.051220282912254333

Shapiro-wilk test of log(density), P values = : 0.5660613775253296

Inference

The p value of log(density) is higher. Therefore, the null hypothesis (that the data is drawn from normal distribution cannot be rejected.

#QUESTION-2

```
import numpy as np
import csv
from scipy import stats
import pandas as pd


datContent = [i.strip().split() for i in open('C:\\Users\\Heera
Baiju\\Downloads\\Data Science\\Assignments\\Assignment 5\\Question 2 data
dat.txt').readlines()]
with open("./HIP_star.csv", "w") as f:
    writer = csv.writer(f)
    writer.writerows(datContent)

data = pd.read_csv('HIP_star.csv')

hyades = data[data['RA']>50]
hyades = hyades[hyades['RA']<100]
hyades = hyades[hyades['DE']>0]
hyades = hyades[hyades['DE']<25]
hyades = hyades[hyades['pmRA']>90]
hyades = hyades[hyades['pmRA']<130]
hyades = hyades[hyades['pmDE']>-60]
hyades = hyades[hyades['pmDE']<-10]
hyades = hyades[hyades['e_Plx']<5]
hyades = hyades[hyades['B-V']<0.2]

df = pd.concat([data, hyades])
non_hyades = df.drop_duplicates(keep = False)

d1 = hyades['B-V'].values
d2 = non_hyades['B-V'].values
d2 = d2[~np.isnan(d2)]

a = np.var(d1)
b = np.var(d2)
print("Hyades color array variance is :", a)
print("Non-hyades color array variance is :", b)


Tstat, pvalue = stats.ttest_ind(d1, d2, equal_var = False)
print("T-statistic value = ",Tstat)
print("2 sample t-test p value = : ",pvalue)
```

OUTPUT

Hyades color array variance is : 0.001848

Non-hyades color array variance is : 0.10768933532979119

T-statistic value = -30.467874175004038

2 sample t-test p value = : 6.291256969608912e-08

Inference

1.We have unequal number of samples and variance. Therefore, ttest_ind from scipy stats with unequal variances was used.

2. The p value is less than 0.05, therefore the null hypothesis can be rejected. (Null hypothesis = that the colors of hyades and non-hyades stars are same). The two star don't have the same colour.