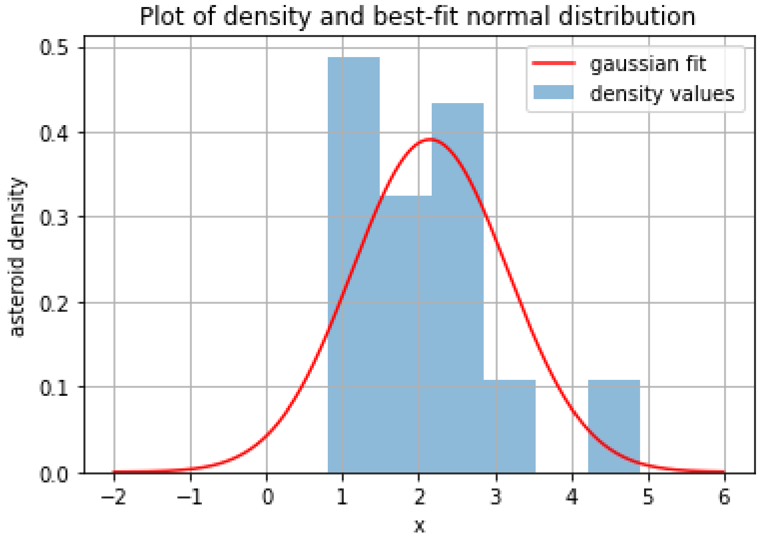
In [1]:

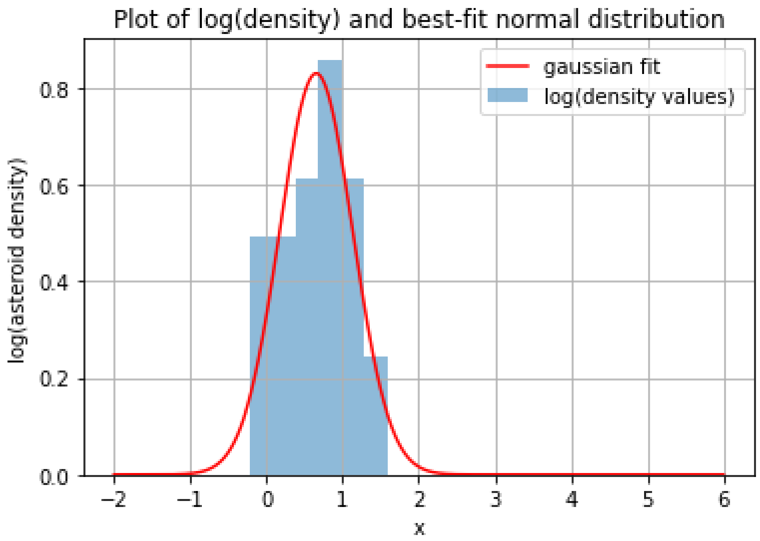
```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
```

In [2]:

```
# given data
data = np.array([[1, 2.12, 0.04], [2, 2.71, 0.11], [4, 3.44, 0.12],
                 [10, 2.76, 1.20], [11, 2.72, 0.12], [15, 0.96, 0.30],
                 [16, 2.00, 0.60], [20, 3.26, 0.60], [22, 2.50, 0.30],
                 [45, 1.20, 0.40], [87, 1.62, 0.30], [90, 1.30, 0.0],
                 [121, 1.96, 0.34], [243, 2.60, 0.50], [253, 1.30, 0.20],
                 [433, 2.67, 0.03], [704, 4.40, 2.10], [762, 1.80, 0.80],
                 [804, 4.90, 3.90], [1999, 2.39, 0.90], [2000, 1.62, 1.05],
                 [2000, 1.47, 0.95], [854, 0.89, 0.13], [1089, 2.52, 0.30],
                 [1313, 1.21, 0.25], [4492, 0.90, 0.10], [617, 0.80, 0.15]])

# density values
density_val = data[:, 1]

#shapiro-wilk test to density values
statistics1, pvalue1 = stats.shapiro(density_val)
print('shapiro-wilk test to density values: \n', 'W =', statistics1, 'p-value =', pvalu
e1)

#shapiro-wilk test to natural log of density values
statistics2, pvalue2 = stats.shapiro(np.log(density_val))
print('shapiro-wilk test to natural log of density values: \n', 'W =', statistics2, 'p-
value =', pvalue2)

#density values plot with gaussian fit
mu1, sigma1 = stats.norm.fit(density_val)

#log of density values with gaussian fit
mu2, sigma2 = stats.norm.fit(np.log(density_val))

# declaring points on x-axis
x = np.linspace(-2,6,1000)

# best-fit gaussian distribution
pdf1 = stats.norm.pdf(x, mu1, sigma1)
pdf2 = stats.norm.pdf(x, mu2, sigma2)

# plot 1
plt.figure()
plt.hist(density_val, density = True, bins='auto', alpha = 0.5, label='density values')
plt.plot(x, pdf1, label='gaussian fit', color='r')
plt.title('Plot of density and best-fit normal distribution')
plt.xlabel('x')
plt.ylabel('asteroid density')
plt.legend(loc='best')
plt.grid()
plt.show()

# plot 2
plt.figure()
plt.hist(np.log(density_val), density = True, bins='auto', alpha=0.5, label='log(densit
y values)')
plt.plot(x, pdf2, label='gaussian fit', color='r')
plt.title('Plot of log(density) and best-fit normal distribution')
plt.xlabel('x')
plt.ylabel('log(asteroid density)')
plt.legend(loc='best')
plt.grid()
plt.show()
```

```
shapiro-wilk test to density values:
 W = 0.9246721863746643 p-value = 0.051220282912254333
shapiro-wilk test to natural log of density values:
 W = 0.9686306715011597 p-value = 0.5660613775253296
```





Clearly, the distribution of natural log of density values is closer to a Gaussian Distribution

Question 2

In [3]:

```
!wget https://iith.ac.in/~shantanud/HIP_star.dat
```

```
--2022-02-21 09:42:31--  https://iith.ac.in/~shantanud/HIP_star.dat
Resolving iith.ac.in (iith.ac.in)... 103.232.241.56
Connecting to iith.ac.in (iith.ac.in)|103.232.241.56|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 170944 (167K)
Saving to: 'HIP_star.dat'

HIP_star.dat        100%[===================>] 166.94K   210KB/s    in 0.8
s

2022-02-21 09:42:34 (210 KB/s) - 'HIP_star.dat' saved [170944/170944]
```

In [4]:

```
data = np.loadtxt('HIP_star.dat', skiprows = 1)

hyades = []
non_hyades = []

for i in range(len(data)):
    if data[i][2] >= 50 and data[i][2] <= 100 and data[i][3] >= 0 and data[i][3] <= 25
and data[i][5] >= 90 and data[i][5] <= 130 and data[i][6] >= -60 and data[i][6] <= -10:
        hyades += [data[i][8]]
    else:
        non_hyades += [data[i][8]]

# converting to numpy array
hyades = np.array(hyades)
non_hyades = np.array(non_hyades)

# finding and printing p-value
statistics = stats.ttest_ind(hyades, non_hyades)
print('two-sample t-test : p-value =',statistics.pvalue)
```

```
two-sample t-test : p-value = 0.00011582222192442334
```

This small value of p-value, assuming null hypothesis : "The Hyades stars have the same color as the non-Hyades stars", indicates that the color of Hyades stars differs from non-Hyades stars