Data Science Analysis Assignment - 5

In [1]:

```
#importing required libraries
import pandas as pd
import math
from scipy import stats
from matplotlib import pyplot as plt
import numpy as np
```

Q1.

In [2]:

```
#Reading asteriod datasheet
df = pd.read_csv("D:\CLASSES\SEM 4\Data Science Analysis\A5\data.csv",sep="\s+")

density = df['Dens']
error = df['+/-']

ln_density = [(math.log(x)) for x in list(density.values)]

shapiro_test = stats.shapiro(density)

w1 = shapiro_test.statistic
p1 = shapiro_test.pvalue

shapiro_test_ln = stats.shapiro(ln_density)

w2 = shapiro_test_ln.statistic
p2 = shapiro_test_ln.pvalue

print("p-value for shapiro test on densities       = %s \np-value for shapiro test on ln of densities = %s" %(p1,p2))
```

```
p-value for shapiro test on densities       = 0.051220282912254333
p-value for shapiro test on ln of densities = 0.5660613775253296
```

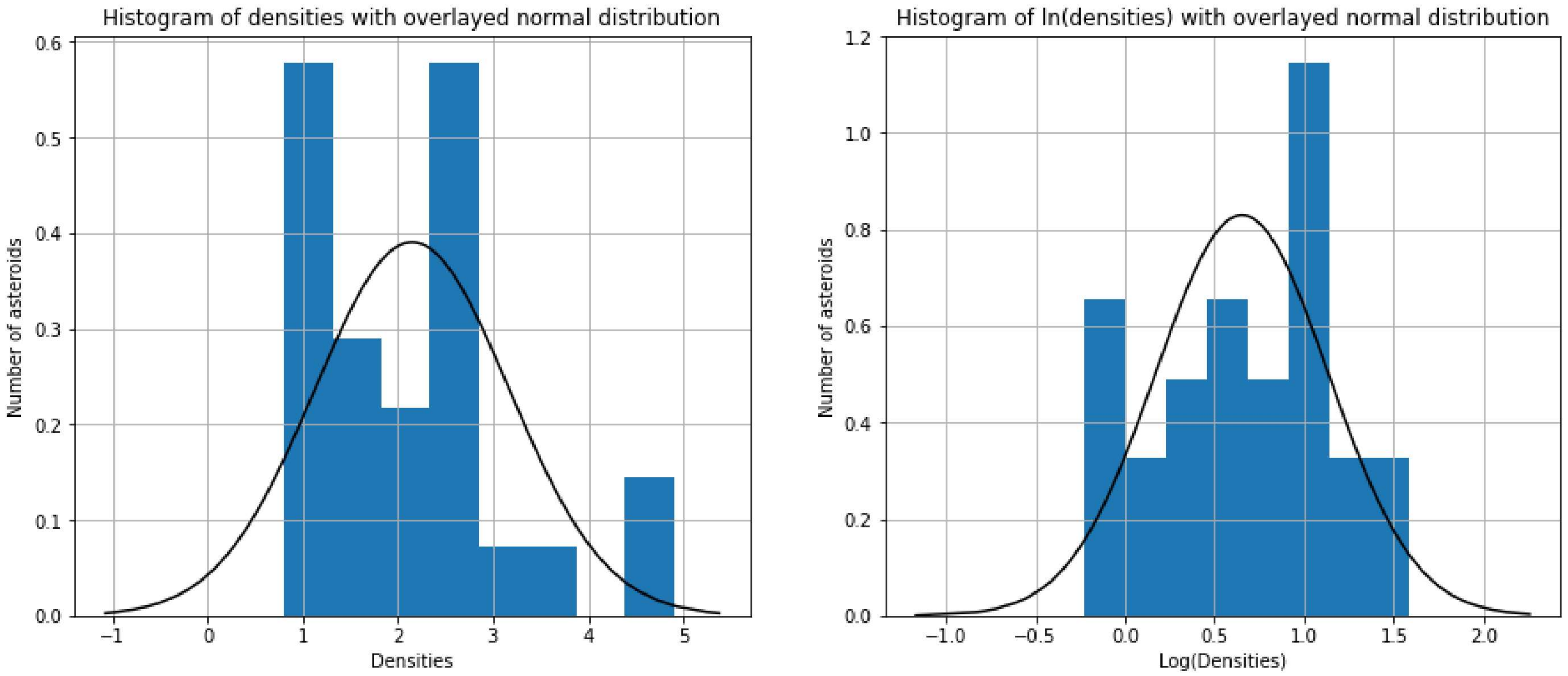
P-value of Shapiro-Wilk test helps us find if we can reject the null-hypothesis. wkt, the null hypothesis says that the data comes from a normal distribution.

Here, p-value for shapiro test on ln of densities is much higher than 0.05. Hence, we can't reject the null hypothesis here. Thus, we don't have sufficient evidence that the data doesn't follow a normal distribution.

Ln of densities is much closer to a normal distribution.

In [3]:

```
#Verifying the above conclusion by plotting histograms of both density and irs Logarithm and overlaying the
#best-fit normal distribution

fig, ax = plt.subplots(1,2,figsize=(15,6))

#densities
mean1,std1 = stats.norm.fit(density)
dist1 = stats.norm(mean1, std1)

x1 = np.sort(dist1.rvs(1000))

ax[0].hist(density, bins=8, density=True)
ax[0].plot(x1,dist1.pdf(x1),'black')

ax[0].set_ylabel('Number of asteroids')
ax[0].set_xlabel('Densities')
ax[0].set_title('Histogram of densities with overlayed normal distribution')
ax[0].grid()

#logarithm of densities
mean2,std2 = stats.norm.fit(ln_density)
dist2 = stats.norm(mean2,std2)

x2 = np.sort(dist2.rvs(1000))

ax[1].hist(ln_density,bins=8,density=True)
ax[1].plot(x2,dist2.pdf(x2),'black')

ax[1].set_xlabel('Log(Densities)')
ax[1].set_ylabel('Number of asteroids')
ax[1].set_title('Histogram of ln(densities) with overlayed normal distribution')
ax[1].grid()
```



Q2.

In [4]:

```
#reading the datasheet
df = pd.read_csv("D:\CLASSES\SEM 4\Data Science Analysis\A5\data2.csv",sep="\s+",usecols=['RA','DE','pmRA','pmDE','B-V'])

#creating 1d arrays
df['RA'] = pd.Series([float(x) for x in list(df['RA'].values)])
df['DE'] = pd.Series([float(x) for x in list(df['DE'].values)])
df['pmRA'] = pd.Series([float(x) for x in list(df['pmRA'].values)])
df['pmDE'] = pd.Series([float(x) for x in list(df['pmDE'].values)])
df['B-V'] = pd.Series([float(x) for x in list(df['B-V'].values)])

#sorting out hyades stars
hyades = df[(50 < df['RA']) & (df['RA'] < 100) & (0 < df['DE']) & (df['DE'] < 25) & (90 < df['pmRA']) &
            (df['pmRA'] < 130) & (-60 < df['pmDE']) & (df['pmDE'] < -10)]
color_hyades = hyades['B-V']

#sorting out non-hyades stars
```

```
non_hyades = pd.concat([df,hyades]).drop_duplicates(keep=False)
color_non_hyades = non_hyades['B-V']

#performing two-sample t-test
t,p = stats.ttest_ind(color_non_hyades,color_hyades)
t,p
```

```
Out[4]: (3.860436921860911, 0.00011582222192442334)
```

As p-value is smaller than 0.05, we can conclude that the colours of hyades and non-hyades stars are different.