```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import shapiro
from scipy.optimize import curve_fit as optimizer

fname= "astro12.csv"
data = (np.loadtxt(fname, delimiter = ',', comments='#', usecols=(0,)))
alpha = 0.05
parameter_names = ["C1", "C2", "Sigma1", "Sigma2", "Mean1", "Mean2"]
def two_gaussian(x,c1,c2,sgm1,sgm2,mean1,mean2):
    part1 = np.abs(c1)/(np.sqrt(2*np.pi*(sgm1**2)))
    part2 = -1*(((x-mean1)**2)/(2*(sgm1**2)))
    part3 = np.abs(c2)/(np.sqrt(2*np.pi*(sgm2**2)))
    part4 = -1*(((x-mean2)**2)/(2*(sgm2**2)))
    first_gauss = part1*np.exp(part2)
    second_gauss = part3*np.exp(part4)
    return first_gauss + second_gauss

n, bins, patches = plt.hist(data, bins=150, zorder=1, color='b')
centres = bins[:-1]+0.5*(bins[1:]-bins[:-1])
plt.title("Histogram of random given values")
plt.xlabel("Given Values")
plt.ylabel("Frequency in distribution")
plt.show()
given, prob_value = shapiro(data)
print("Normality test p-value (Shapiro-Wilk) = {:.3e}".format(prob_value))
print("Given alpha (for comparison): {}".format(alpha))
if prob_value<alpha:
    print("Data is not Gaussian as it has failed the Normality test")
else:
    print("Data is Gaussian")

starting_guess = [11500,4000, 3, 3, 0, -3]
plt.hist(data, bins=150, color='b', zorder=1)
plt.plot(centres, two_gaussian(centres, *starting_guess), 'r--', zorder=5, linewidth=3)
plt.title("Histogram with starting fit for given data")
plt.xlabel("Given Values")
plt.ylabel("Frequency in distribution")
plt.show()
plt.hist(data, bins=150, color='b', zorder=1)
parameters, coverance = optimizer(two_gaussian, centres, n, p0=starting_guess)
plt.plot(centres, two_gaussian(centres, *parameters), 'r--', zorder=5, linewidth=3)
plt.title("Fitted Gaussian curves with Given values")
plt.xlabel("Given Values")
plt.ylabel("Frequency in Distribution")
plt.show()
print("The fitted paramaters are as such \n")
for i in range(len(parameters)):
    print ("{} = {:.4}".format(parameter_names[i], parameters[i]))
```