Demarcus Cotto

726005570

ASTR 320 – 500

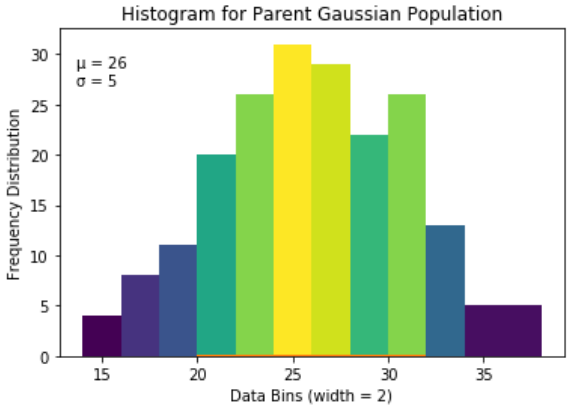
29 March 2020

ASSIGNMENT 3

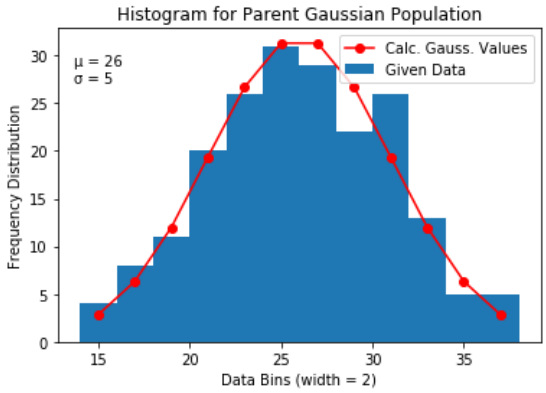
In [191]:
```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import colors
from matplotlib.ticker import PercentFormatter
from scipy import fft
from scipy import signal
```

In [192]:
```
# part a: Plotting the histogram
frequency = [4,8,11,20,26,31,29,22,26,13,5,2,3]
bin = [14,16,18,20,22,24,26,28,30,32,34,36,38]

x = np.zeros(200)
i = 0
for j in range(0,np.size(frequency)):
    for k in range(0,frequency[j]):
        x[i] = bin[j]
        i = i + 1

N, bins, patches = plt.hist(x, bins = [14,16,18,20,22,24,26,28,30,32,34,36,38])

fracs = N / N.max()
norm = colors.Normalize(fracs.min(),fracs.max())

for thisfrac, thispatch in zip(fracs, patches):
    color = plt.cm.viridis(norm(thisfrac))
    thispatch.set_facecolor(color)


plt.figure(1)
plt.hist(x, bins = [14,16,18,20,22,24,26,28,30,32,34,36,38],density = True)
plt.ylabel('Frequency Distribution')
plt.xlabel('Data Bins (width = 2)')
plt.title('Histogram for Parent Gaussian Population')
plt.figtext(.150,.75,"\u03BC = 26 \n\u03C3 = 5")
```

Out[192]: Text(0.15, 0.75, 'μ = 26 \nσ = 5')



In [198]:
```
# part b:
mu = 26
std = 5
bin_new = np.zeros(np.size(bin) -1)
gauss_func = np.zeros(np.size(bin_new))

for  i in range(0,np.size(bin) - 1):
    bin_new[i] = bin[i] + 1
    gauss_func[i] = (1/(std*np.sqrt(2*np.pi))) * np.exp((-1/2)*(bin_new[i] - mu)**2/std**2) * 400 # to normalize to the area of
the histogram - multiply by 400

plt.figure(2)
plt.hist(x, bins = [14,16,18,20,22,24,26,28,30,32,34,36,38], label = 'Given Data')
plt.ylabel('Frequency Distribution')
plt.xlabel('Data Bins (width = 2)')
plt.title('Histogram for Parent Gaussian Population')
plt.figtext(.150,.75,"\u03BC = 26 \n\u03C3 = 5")
plt.plot(bin_new,gauss_func,'o-', color = 'red', label = 'Calc. Gauss. Values')
plt.legend(loc = 'upper right')
plt.show()

print(gauss_func)
```




```
[ 2.83796743  6.31601266 11.97819725 19.35765796 26.65796823 31.28341552
 31.28341552 26.65796823 19.35765796 11.97819725  6.31601266  2.83796743]
```

In [194]:
```
# part c:
chi_2 = np.zeros(np.size(x))
x_mid = np.zeros(np.size(x))

for i in range(0, np.size(x)):
    x_mid[i] = x[i] + 1
    chi_2[i] = ((x_mid[i] - mu) / std)**2

chi_2_sum = np.sum(chi_2)
print(chi_2_sum)
```

```
204.15999999999997
```