NGANGA CAROLINE KANUTHU

BSC IN COMPUTER SCIENCE

SCT211-0020/2017

SCIENTIFIC COMPUTING

ASSIGNMENT LAB 01

In [15]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

In [16]:
```
#displaying my dataset
c=pd.read_csv('all-ages.csv')
c
```

Out[16]:

	Major_code	Major	Major_category	Total	Employed	Employed_full_time_year_round	Une
0	1100	GENERAL AGRICULTURE	Agriculture & Natural Resources	128148	90245	74078	
1	1101	AGRICULTURE PRODUCTION AND MANAGEMENT	Agriculture & Natural Resources	95326	76865	64240	
2	1102	AGRICULTURAL ECONOMICS	Agriculture & Natural Resources	33955	26321	22810	
3	1103	ANIMAL SCIENCES	Agriculture & Natural Resources	103549	81177	64937	
4	1104	FOOD SCIENCE	Agriculture & Natural Resources	24280	17281	12722	
5	1105	PLANT SCIENCE AND AGRONOMY	Agriculture & Natural	79409	63043	51077	

In [17]:
```
#slicing data to reduce the number of entries to 100
CROPPED_DATA=c.head(100)
CROPPED_DATA
```

Out[17]:

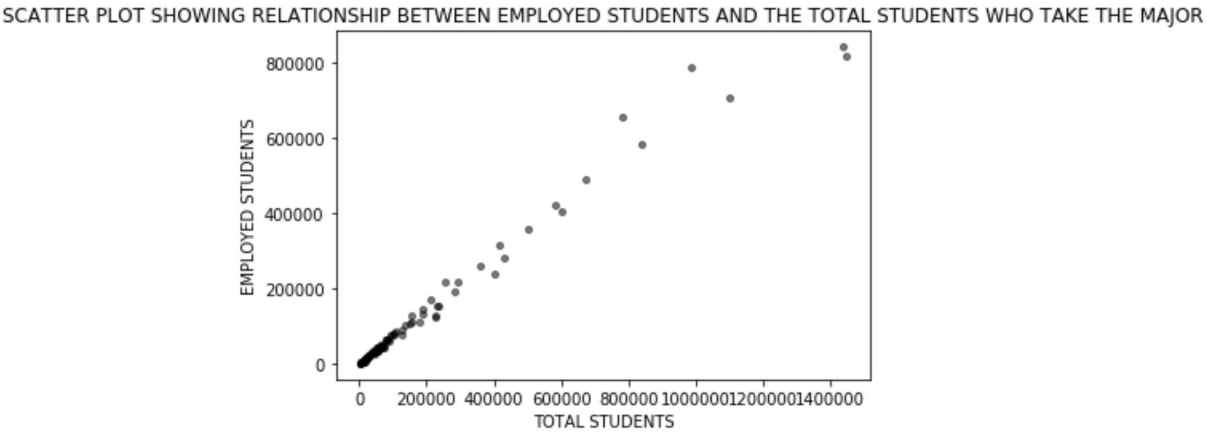
	Major_code	Major	Major_category	Total	Employed	Employed_full_time_year_round
0	1100	GENERAL AGRICULTURE	Agriculture & Natural Resources	128148	90245	74078
1	1101	AGRICULTURE PRODUCTION AND MANAGEMENT	Agriculture & Natural Resources	95326	76865	64240
2	1102	AGRICULTURAL ECONOMICS	Agriculture & Natural Resources	33955	26321	22810
3	1103	ANIMAL SCIENCES	Agriculture & Natural Resources	103549	81177	64937
4	1104	FOOD SCIENCE	Agriculture & Natural Resources	24280	17281	12722
5	1105	PLANT SCIENCE AND AGRONOMY	Agriculture & Natural Resources	79409	63043	51077
6	1106	SOIL SCIENCE	Agriculture & Natural Resources	6586	4926	4042
7	1199	MISCELLANEOUS AGRICULTURE	Agriculture & Natural Resources	8549	6392	5074
8	1301	ENVIRONMENTAL SCIENCE	Biology & Life Science	106106	87602	65238
9	1302	FORESTRY	Agriculture & Natural Resources	69447	48228	39613
10	1303	NATURAL RESOURCES MANAGEMENT	Agriculture & Natural Resources	83188	65937	50595
11	1401	ARCHITECTURE	Engineering	294692	216770	163020
12	1501	AREA ETHNIC AND CIVILIZATION STUDIES	Humanities & Liberal Arts	103740	75798	50530
13	1901	COMMUNICATIONS	Communications & Journalism	987676	790696	595739
14	1902	JOURNALISM	Communications & Journalism	418104	314438	235407
15	1903	MASS MEDIA	Communications & Journalism	211213	170474	125489
16	1904	ADVERTISING AND PUBLIC RELATIONS	Communications & Journalism	186829	147433	111552
17	2001	COMMUNICATION TECHNOLOGIES	Computers & Mathematics	62141	49609	37261
18	2100	COMPUTER AND INFORMATION SYSTEMS	Computers & Mathematics	253782	218248	189950
19	2101	COMPUTER PROGRAMMING AND DATA PROCESSING	Computers & Mathematics	29317	22828	18747
20	2102	COMPUTER SCIENCE	Computers & Mathematics	783292	656372	561052
21	2105	INFORMATION SCIENCES	Computers & Mathematics	77805	66393	57604

SCATTER PLOT

In [20]:
```
#SCATTER PLOT two display the relationship between the employed students and the numb
x=CROPPED_DATA.Total
y=CROPPED_DATA.Employed
colors=(0,0,0)
area = np.pi*5
plt.scatter(x, y, s=area, c=colors, alpha=0.5)
plt.title('SCATTER PLOT SHOWING RELATIONSHIP BETWEEN EMPLOYED STUDENTS AND THE TOTAL
plt.xlabel('TOTAL STUDENTS')
plt.ylabel('EMPLOYED STUDENTS')

plt.show()
```

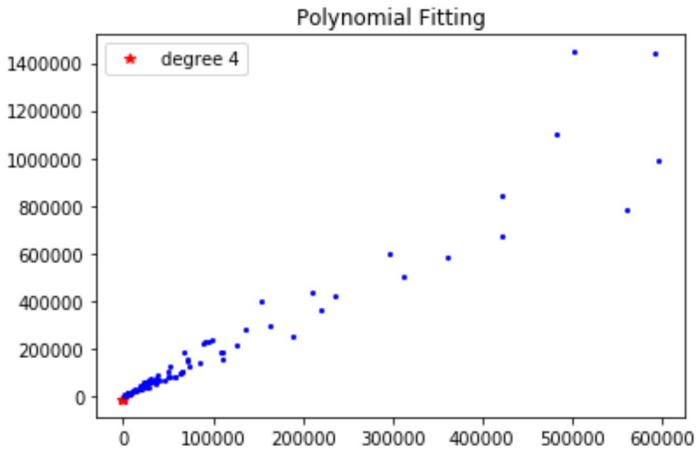
```
'c' argument looks like a single numeric RGB or RGBA sequence, which should be avo
ided as value-mapping will have precedence in case its length matches with 'x' & '
y'.  Please use a 2-D array with a single row if you really want to specify the sa
me RGB or RGBA value for all points.
```



POLYNOMIAL FITS

In [122]:
```
#plotting polynomial fits
x=CROPPED_DATA['Employed_full_time_year_round'].values
y=CROPPED_DATA['Total'].values
#plot the polynomials and points
p4=np.polyfit(x,y,4)
p4=np.polyfit(x,y,4)
#creating an abstraction for the math operation
xp4=np.poly1d(p4)
xp2=np.poly1d(p4)
xx=np.linspace(0,100,4)
#plot the polynomials and points
plt.plot(xx,np.polyval(p4,xx),"r*",label='degree 4')
plt.plot(x,y,'bo',markersize=2)
plt.legend(loc='upper left')
plt.title('Polynomial Fitting')
```

Out[122]:
```
Text(0.5, 1.0, 'Polynomial Fitting')
```

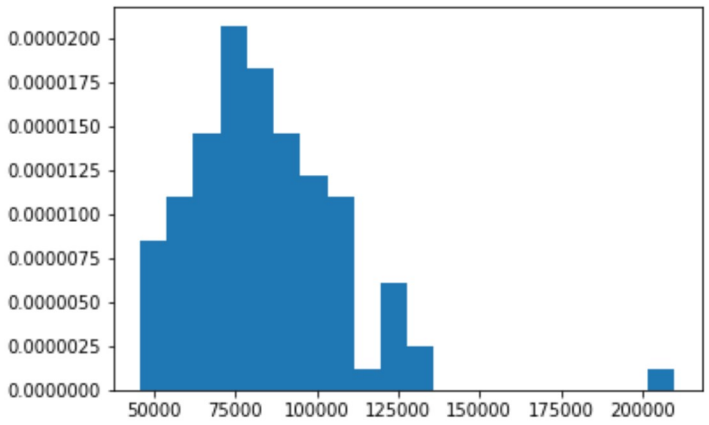


HISTOGRAM

In [36]:
```
#plotting a histogram using the hist()function
x=CROPPED_DATA['P25th'].values
y=CROPPED_DATA['P75th'].values
sorted_x=np.sort(x)
mu=np.mean(sorted_x)
sigma=np.std(sorted_x)
X=mu+sigma*x
bins=15
fig, ax = plt.subplots()
# the histogram of the data

plt.hist(y,20,normed=1)
```

Out[36]:
```
(array([8.52618758e-06, 1.09622412e-05, 1.46163216e-05, 2.07064555e-05,
        1.82704019e-05, 1.46163216e-05, 1.21802680e-05, 1.09622412e-05,
        1.21802680e-06, 6.09013398e-06, 2.43605359e-06, 0.00000000e+00,
        0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
        0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 1.21802680e-06]),
 array([ 45800.,  54010.,  62220.,  70430.,  78640.,  86850.,  95060.,
        103270., 111480., 119690., 127900., 136110., 144320., 152530.,
        160740., 168950., 177160., 185370., 193580., 201790., 210000.]),
 <a list of 20 Patch objects>)
```



NORMAL DISTRIBUTION

In [103]:

```
mu=np.mean(x)
sigma=np.std(x)
x_array=np.linspace(0,100,100)
plt.plot(x_array,(1/np.sqrt(2*np.pi*sigma**2))*np.exp(-(x_array-mu)**2/(2*sigma**2)),
plt.xlabel('')
plt.ylabel('')
plt.title('Histogram with $\mu={}, \quad\sigma={}$'.format(mu,sigma))
# Tweak spacing to prevent clipping of ylabel
fig.tight_layout()
plt.show()
```