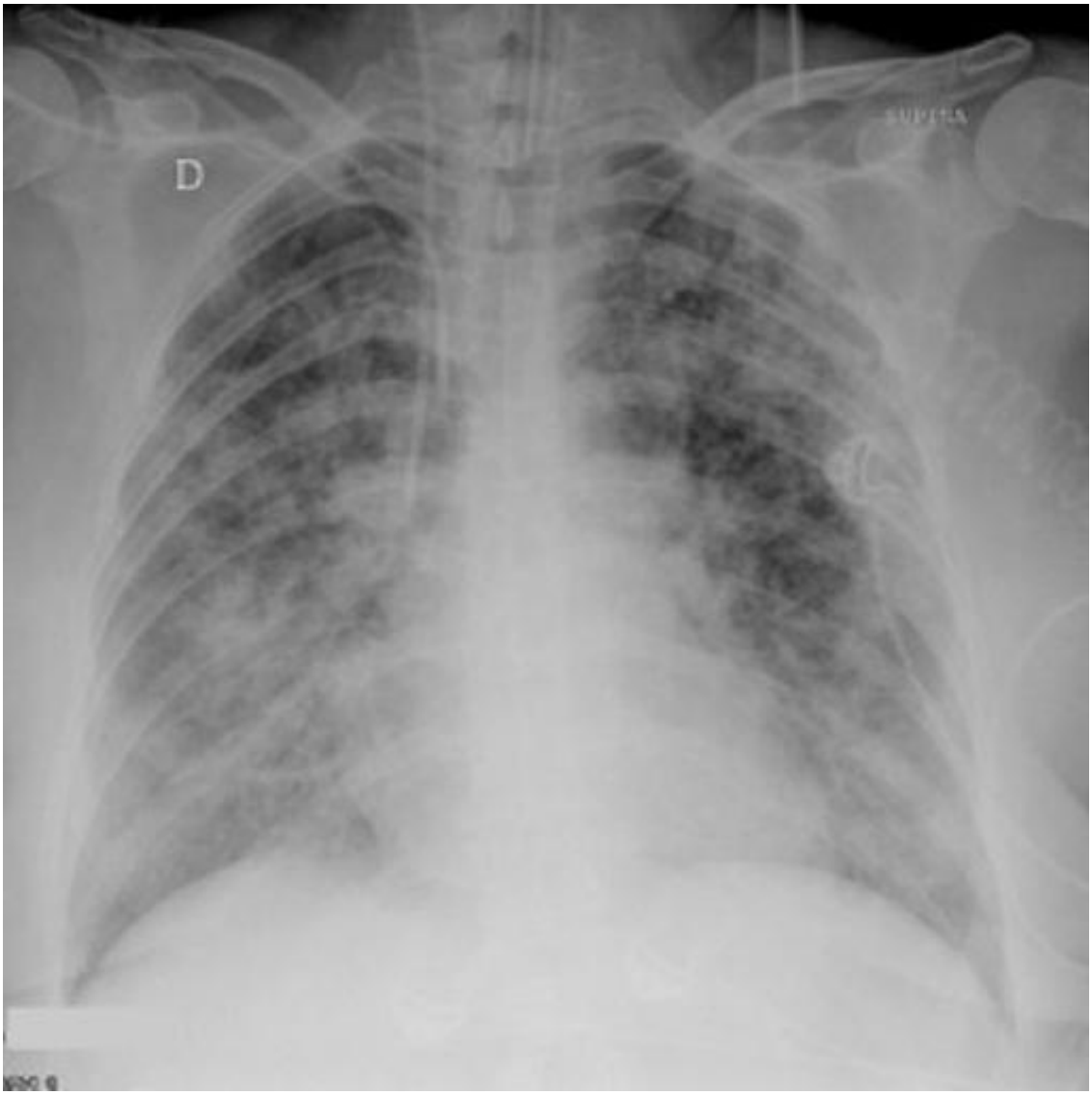
Histogram Equalization

December 13, 2019

[1]:
```
from IPython.display import display, Math, Latex
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image, ImageFilter
import os, os.path
import glob
from copy import deepcopy, copy
```

[2]:
```
ima = []
for filename in glob.glob('imagesHW7/*.jpg'): #assuming gif
    im = Image.open(filename)
    im = im.convert('L')
    ima.append(im)
```

[3]:
```
for i in range(len(ima)):
    display(ima[i])
```


D

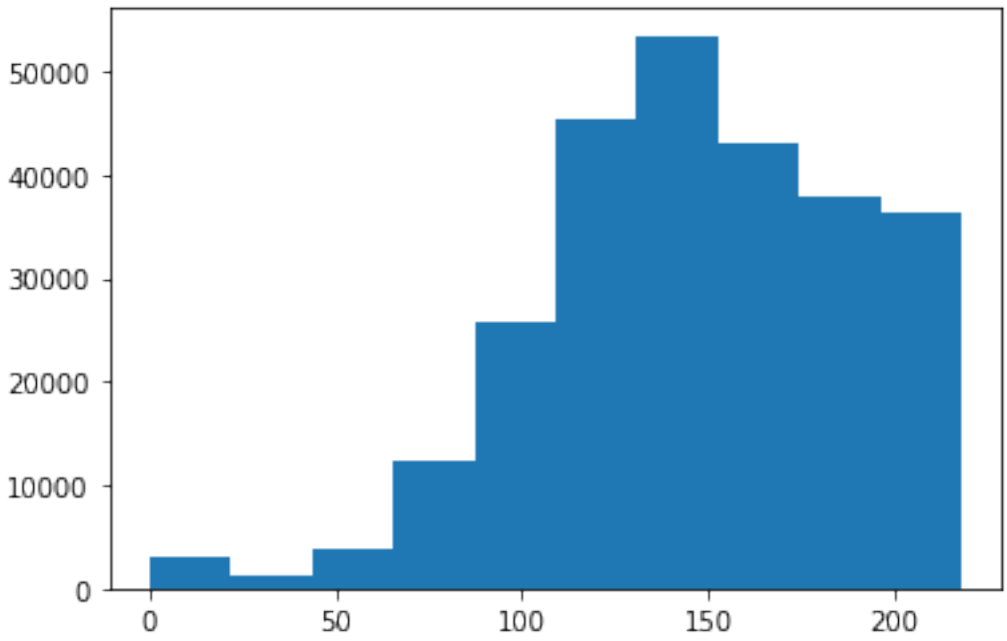
[4]:
```
# for i in range(len(ima)):
#     img = ima[i].resize((512,512), Image.ANTIALIAS)
#     img.save("imagesHW7/image" + str(i+10) + ".jpg")
```

[5]:
```
test = np.asarray(ima[2])
```

[6]:
```
#turns the 2D pixel to a 1D array of pixels
flat = test.flatten()
```

[7]:
```
#image histogram
plt.hist(flat)
```

[7]:
```
(array([ 3065.,  1169.,  3812., 12278., 25750., 45402., 53458., 42982.,
        37942., 36286.]),
 array([  0. ,  21.8,  43.6,  65.4,  87.2, 109. , 130.8, 152.6, 174.4,
        196.2, 218. ]),
 <a list of 10 Patch objects>)
```



[8]:
```
# formula for creating the histogram
display(Math(r'P_x(j) = \sum_{i=0}^{j} P_x(i)'))
test[0].dtype
```

$$P_x(j) = \sum_{i=0}^{j} P_x(i)$$

[8]:
```
dtype('uint8')
```

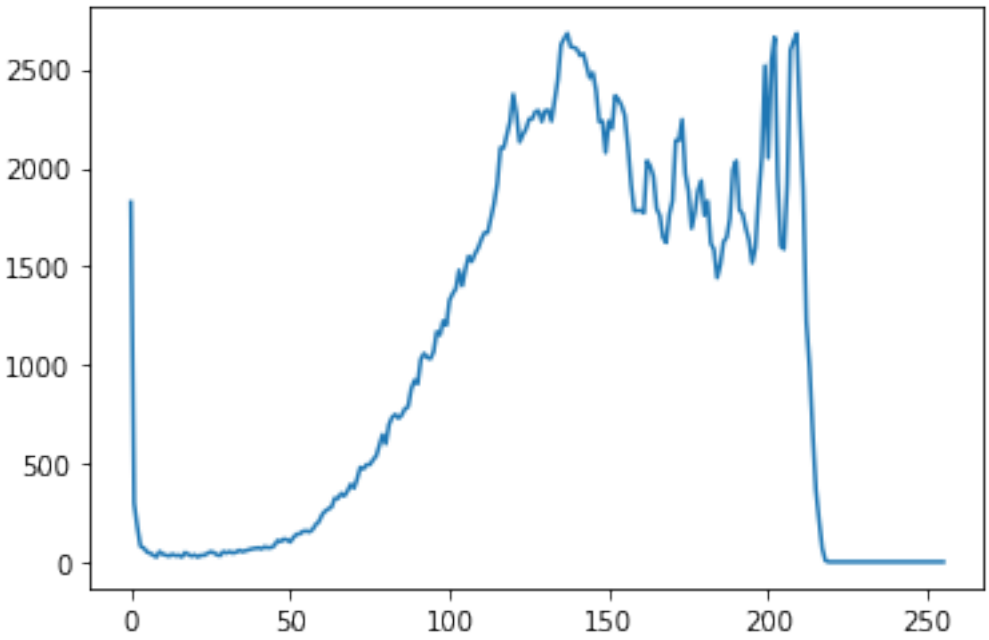
[9]:
```
def create_histogram(image, bins):
    #pixel range ex:1-256
    np_arr = np.asarray(image)
    flat = np_arr.flatten()
    histogram = np.zeros(bins)

    for pixel in flat:
        histogram[pixel] += 1

    return histogram
```

[10]:
```
hist = create_histogram(ima[2], 256)
plt.plot(hist)
```

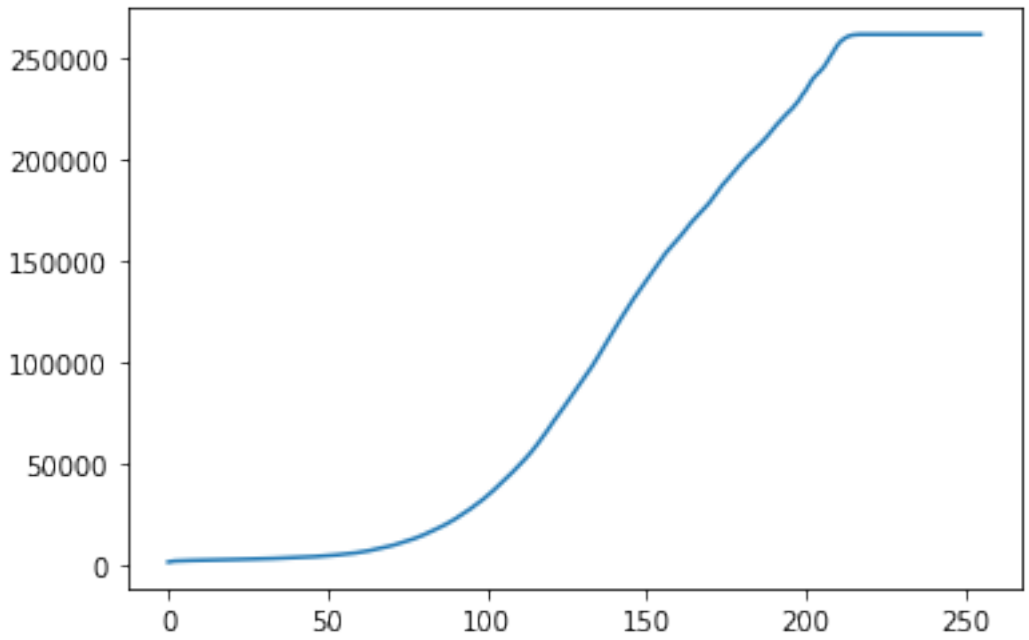
[10]: [<matplotlib.lines.Line2D at 0x1974d1b5b48>]



[11]:
```
def cumulative_sum(histogram):
    histogram = iter(histogram)
    b = [next(histogram)]
    for i in histogram:
        b.append(b[-1] + i)
    return np.array(b)
```

[12]:
```
csum = cumulative_sum(hist)
plt.plot(csum)
```

[12]: [<matplotlib.lines.Line2D at 0x1974d22a448>]



[13]:
```
# formula to calculate cumulation sum
display(Math(r's_k = \sum_{j=0}^{k} {\frac{n_j}{N}}'))
```

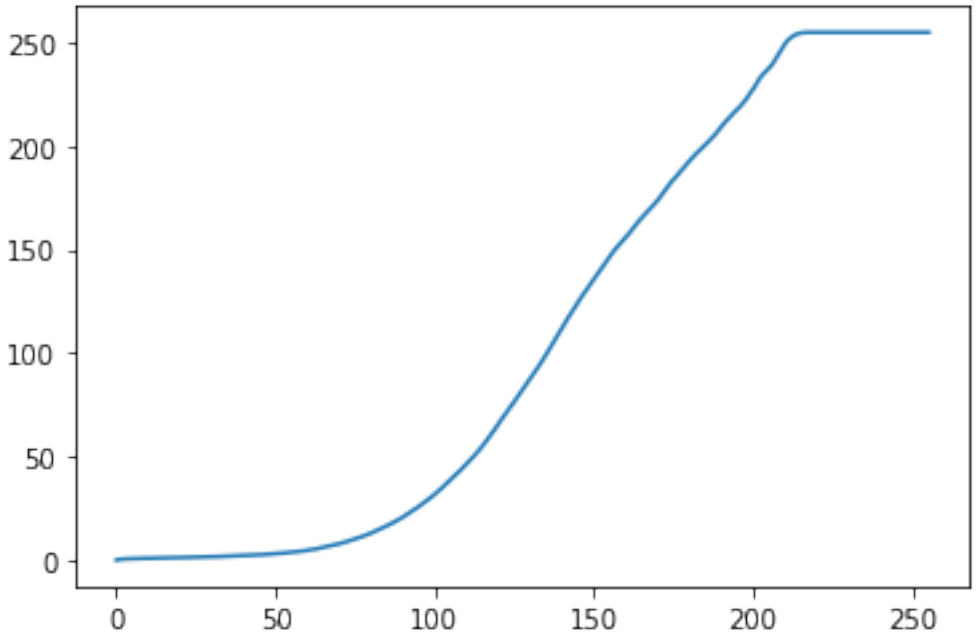
$$s_k = \sum_{j=0}^{k} \frac{n_j}{N}$$

[14]:
```
nj = (csum - csum.min()) * 255
N = csum.max() - csum.min()

csum_normalize = nj/N

plt.plot(csum_normalize)
```

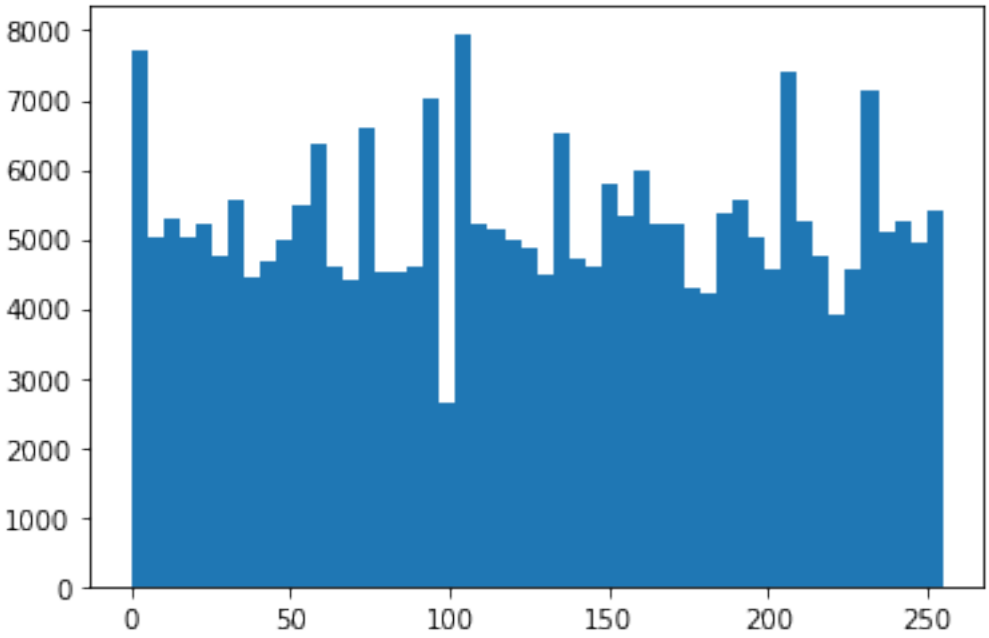
[14]: [<matplotlib.lines.Line2D at 0x1974d294208>]



[15]:
```
csum_normalize = csum_normalize.astype('uint8')
```

[16]:
```
equalize_img = csum_normalize[flat]
plt.hist(equalize_img, bins=50)
```

[16]:
```
(array([7724., 5010., 5281., 5012., 5221., 4742., 5556., 4439., 4682.,
        4980., 5485., 6371., 4608., 4405., 6612., 4535., 4524., 4582.,
        7024., 2625., 7953., 5213., 5149., 4977., 4874., 4471., 6513.,
        4713., 4582., 5805., 5338., 6001., 5198., 5218., 4278., 4215.,
        5350., 5579., 5033., 4574., 7415., 5245., 4746., 3896., 4568.,
        7139., 5105., 5240., 4933., 5405.]),
 array([  0. ,   5.1,  10.2,  15.3,  20.4,  25.5,  30.6,  35.7,  40.8,
         45.9,  51. ,  56.1,  61.2,  66.3,  71.4,  76.5,  81.6,  86.7,
         91.8,  96.9, 102. , 107.1, 112.2, 117.3, 122.4, 127.5, 132.6,
        137.7, 142.8, 147.9, 153. , 158.1, 163.2, 168.3, 173.4, 178.5,
        183.6, 188.7, 193.8, 198.9, 204. , 209.1, 214.2, 219.3, 224.4,
        229.5, 234.6, 239.7, 244.8, 249.9, 255. ]),
 <a list of 50 Patch objects>)
```



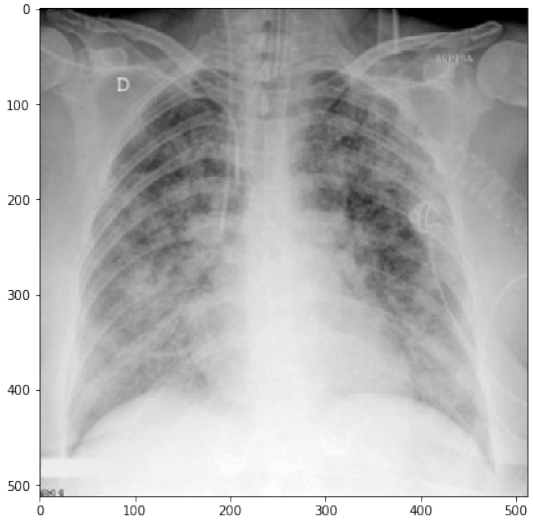
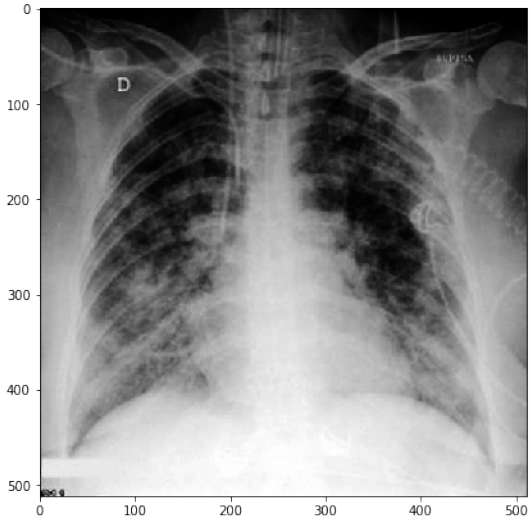
[17]:
```
equalize_img = np.reshape(equalize_img, ima[2].size)
```

[20]:
```
# set up side-by-side image display
fig = plt.figure()
fig.set_figheight(15)
fig.set_figwidth(15)

fig.add_subplot(1,2,1)
plt.imshow(ima[2], cmap='gray')

# display the new image
fig.add_subplot(1,2,2)
plt.imshow(equalize_img, cmap='gray')

plt.show(block=True)
```


0
100
200
300
400
500
D
0
100
200
300
400
500

0
100
200
300
400
500
D
0
100
200
300
400
500