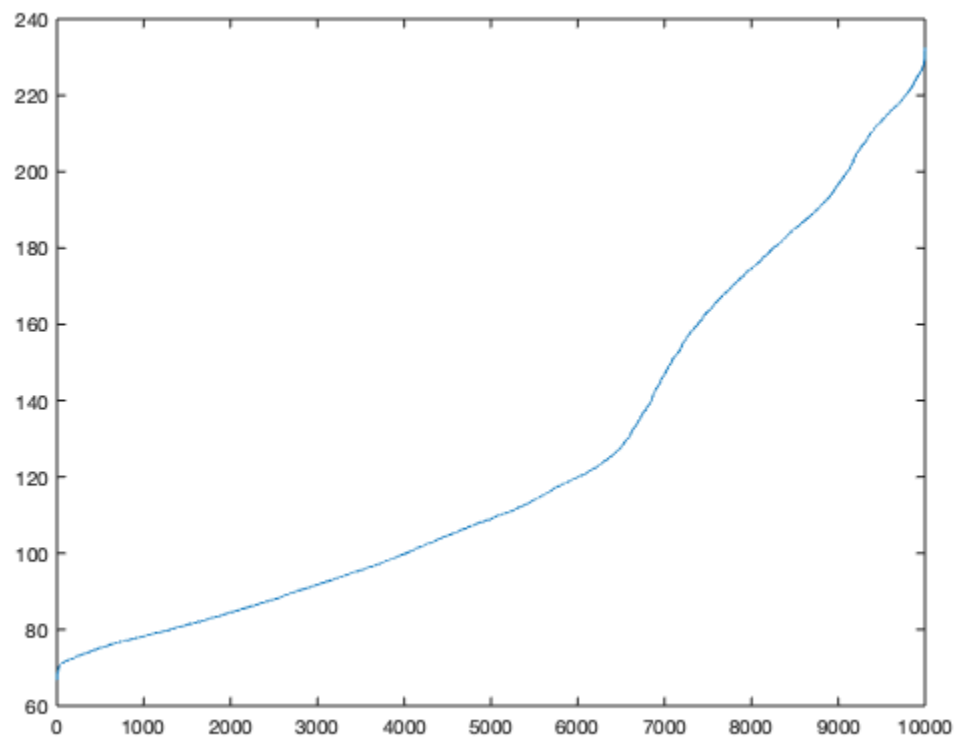

Part 0. Getting Started

```
img = double(imread('../images/flowergray.jpg'));  
A = imresize(img,[100,100]);
```

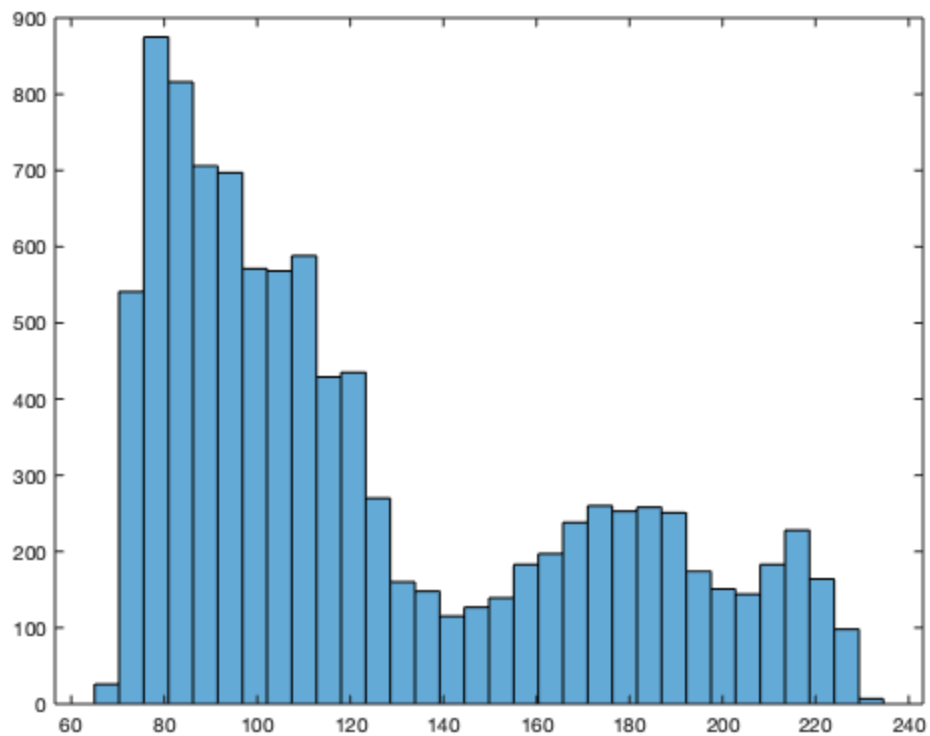
a. Sort all the intensities in A, put the result in a single 10000-dimensional vector x, and plot the values in x.

```
x = sort(A(:));  
figure("Name", "Vector x");  
plot(x)
```



b. Display a figure showing a histogram of A's intensities with 32 bins.

```
figure("Name", "Histogram of A's intensities");  
histogram(A,32);
```



c. Create and display a new binary image with the same size as A, which is white wherever the intensity in A is greater than a threshold t, and black everywhere else.

```
t = 100;  
binaryImage = imbinarize(A, t);  
figure("Name", "Binary image A");  
imshow(binaryImage)
```



d. Generate a new image (matrix), which has the same size as A, but with A's mean intensity value subtracted from each pixel. Set any negative values to 0.

```
meanIntensity = mean(mean(A));  
Aprime = A - meanIntensity;  
Aprime(Aprime < 0) = 0;
```

e. Let y be the vector: y = [1: 8]. Use the reshape command to form a new matrix s whose first column is [1, 2, 3, 4]?, and whose second column is [5, 6, 7, 8]?

```
y = [1:8];  
newY = reshape(y,[4,2]);
```

f. Create a vector [1, 3, 5 ?, 99]. Extract the corresponding pixel from the image in its two dimensions, i.e., subsample the original image to its half size.

```
indexs = (1:2:99);  
downImage = A(indexs,indexs);
```

g. Use fspecial to create a Gaussian Filter and then apply the imfilter function to the image with the created Gaussian Filter, by doing so you should see a blurred image. Change three combinations of parameters of the Gaussian Filter and compare the results.

```
filter1 = fspecial('gaussian',[3,3], 1);  
image1 = imfilter(A, filter1);  
filter2 = fspecial('gaussian',[5,5], 10);  
image2 = imfilter(A, filter2);  
filter3 = fspecial('gaussian',[13,13], 10);  
image3 = imfilter(A, filter3);
```

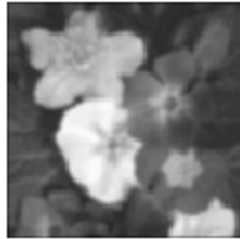
```
figure("Name", "Gaussian filter");
```

```
subplot(131)  
imshow(image1,[])  
title("Filter size [3,3], sigma 1");
```

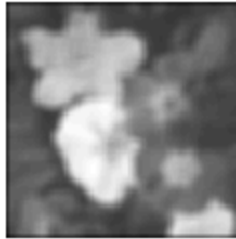
```
subplot(132)  
imshow(image2,[])  
title("Filter size [5,5], sigma 5");
```

```
subplot(133)  
imshow(image3,[])  
title("Filter size [13,13], sigma 10");
```

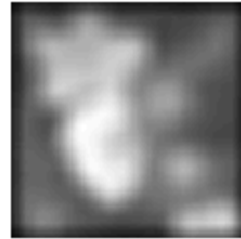
Filter size [3,3], sigma 1



Filter size [5,5], sigma 5

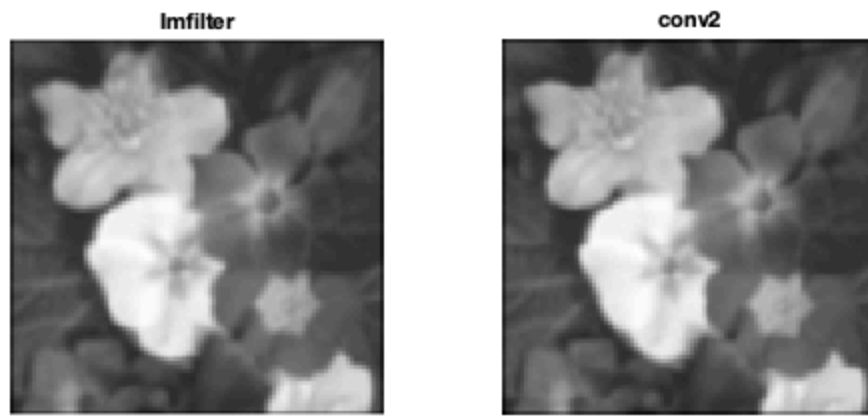


Filter size [13,13], sigma 10



h. Apply the `conv2` instead of `imfilter` function to the same process (for one Gaussian Filter), do you see any changes? Why?

```
conImage1 = conv2(A, filter1, 'same');  
  
figure("Name", "Conv vs imfilter");  
  
subplot(121)  
imshow(image1,[])  
title("Imfilter");  
  
subplot(122)  
imshow(conImage1,[])  
title("conv2");  
% *No changes, they are same*
```



Published with MATLAB® R2019b