Exercicio 1:

Leia a ajuda para as funções imadjuste imhist.

Leia as imagens Einstein_low_contrast.png, Einstein_med_contrast.png e Einstein_high_contrast.png.

```
>> Alb_high = imread("Einstein_high_contrast.png");
>> Alb_med = imread("Einstein_med_contrast.png");
>> Alb_low = imread("Einstein_low_contrast.png");
```

Apresente cada imagem e seu respectivo histograma em uma figura diferente. Compare os histogramas.

```
>> figure(1)
>> imhist(Alb_high)
>> figure(2)
>> imhist(Alb_med)
>> figure(3)
>> imhist(Alb_low)
```

Figura 1:

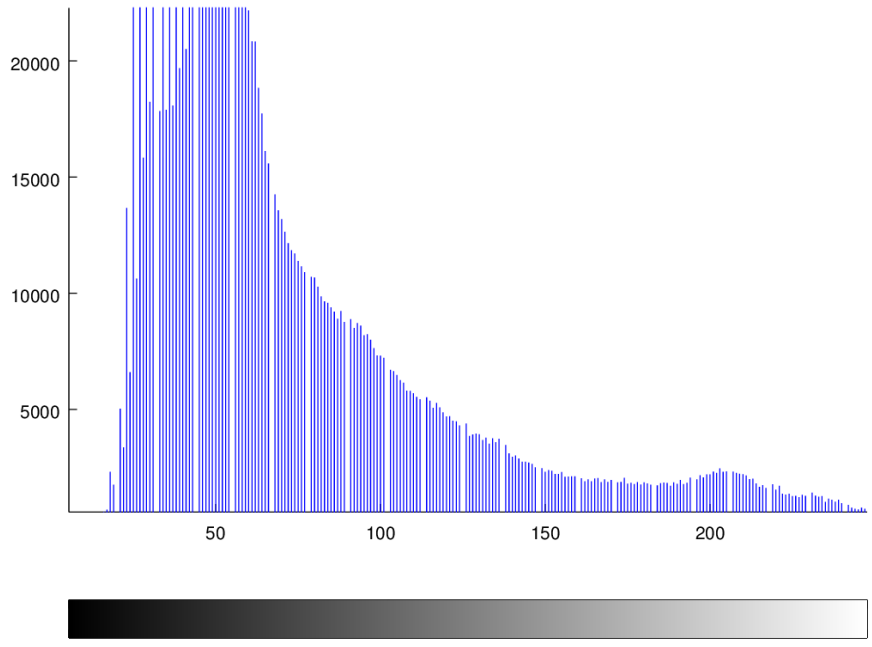


Figura 2:

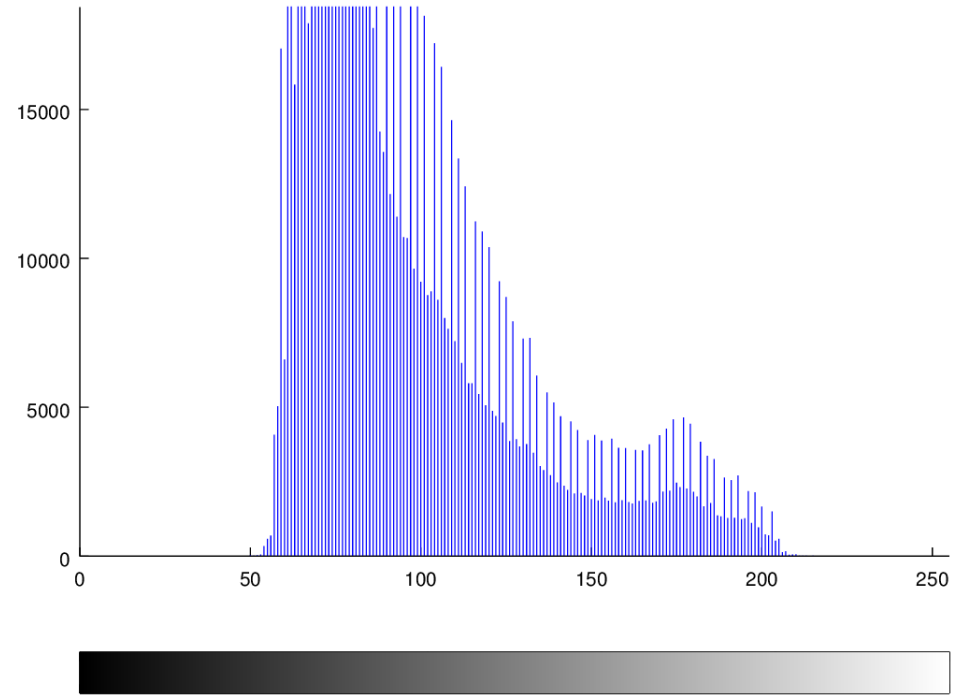
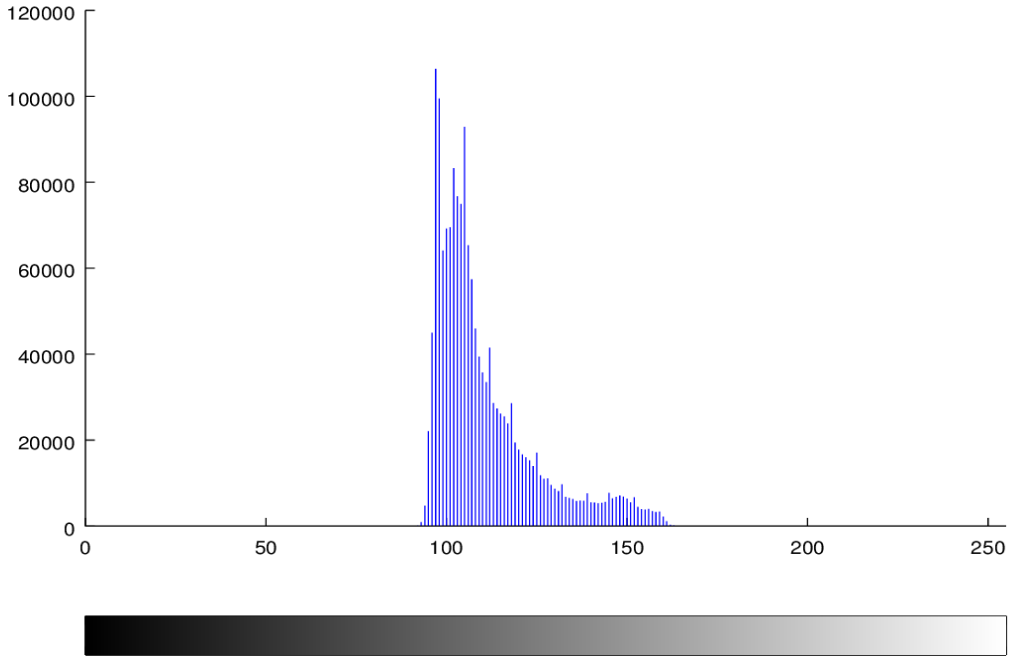

15000
10000
5000
0
0
50
100
150
200
250

Figura 3:

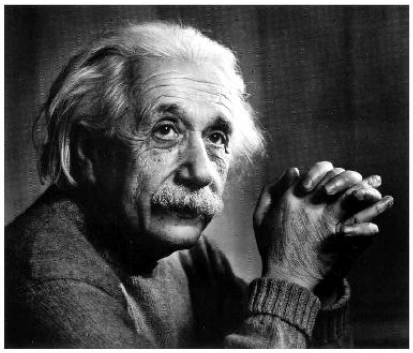

120000
100000
80000
60000
40000
20000
0
0
50
100
150
200
250

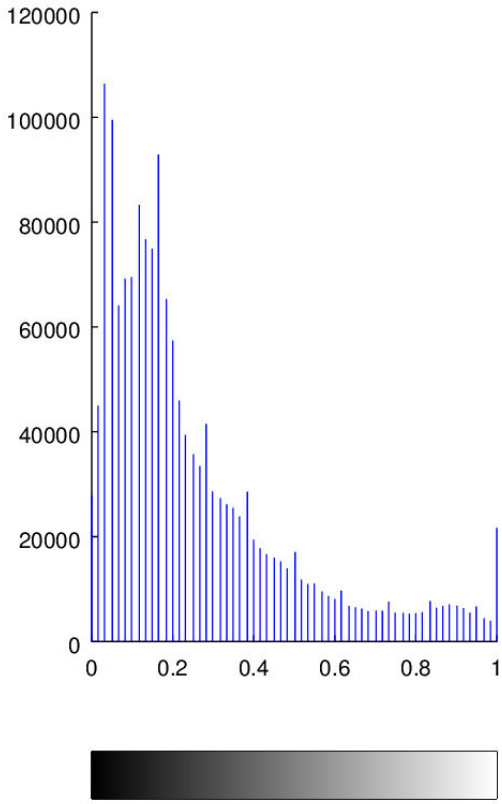
Utilize a função imadjust para melhorar o contraste da imagem Einstein_low_contrast.png, de forma a que o histograma da imagem resultante seja parecido com o da imagem Einstein_high_contrast.png.

```
>> Alb_low_new = im2double(Alb_low);
>> Alb_low_new = imadjust(Alb_low_new)
>> imshow(Alb_low_new)
```

Apresente a nova imagem e seu histograma em uma única figura.

```
>> subplot(1,2,1)
>> imshow(Alb_low_new)
>> subplot(1,2,2)
>> imhist(Alb_low_new);
```





Exercicio 2:

Leia a imagem leme.bmp.

```
>> Leme = imread("leme.bmp")
```

Utilizando a função imadust, crie uma nova imagem
colorida em que as áreas escuras da imagem original fiquem mais claras, mas as áreas claras da imagem original não mudem muito de intensidade na nova imagem.

```
>> Lemea = imadjust(Leme, [0;1],[0;1], 0.45);
>> imshow(Lemea);
```

Apresente a imagem original e a nova imagem em
uma única figura.

```
>> subplot(2,1,1);
>> imshow(Leme);
>> subplot(2,1,2);
>> imshow(Lemea);
```

Exercício 3:

Leia a ajuda para a função imfilter.

Escrever um script que:
a)Lê uma imagem RGB.
b)Cria uma máscara espacial para suavização linear.
c)Filtra a imagem de entrada (todos os planos) com esta máscara.
d)Mostra as imagem original e filtrada numa única figura.

Execute o script para pelo menos duas imagens e apresente os resultados.

```
>> Gato = imread("gato.jpg");
>> Cachorro = imread("caes.jpg");
>> filtro = fspecial("gaussian", 60);
>> Gato_mascara = imfilter(Gato, filtro);
>> Cachorro_mascara = imfilter(Cachorro, filtro);
>> subplot(1,2,1);
```

```
>> imshow(Gato);
>> subplot(1,2,2);
>> imshow(Gato_mascara);
```

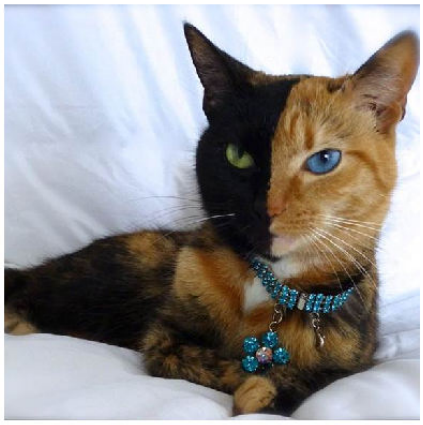
```
>> subplot(1,2,1);
>> imshow(Cachorro);
>> subplot(1,2,2);
>> imshow(Cachorro_mascara);
```

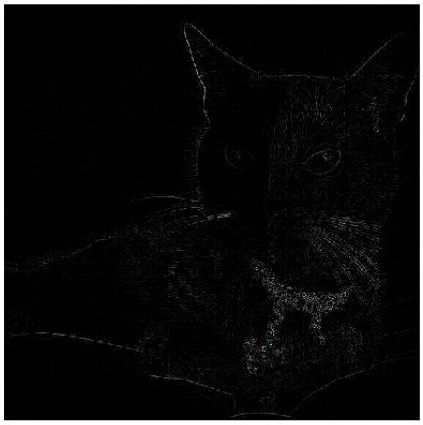
Exercício 4:

Leia a ajuda da função fspecial as máscaras disponíveis para filtragem linear.

Repita o exercício anterior, usando algumas (ao menos 2)
das máscaras disponíveis a partir de fspecial.

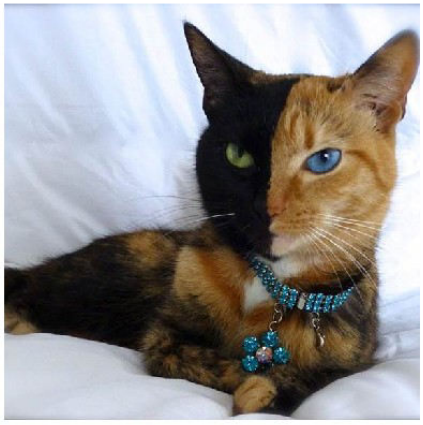
```
>> Gato = imread("gato.jpg");
>> Cachorro = imread("caes.jpg");
>> filtro = fspecial("laplacian", 0.3);
>> Gato_mascara = imfilter(Gato, filtro);
>> Cachorro_mascara = imfilter(Cachorro, filtro);
>> subplot(1,2,1);
>> imshow(Gato);
>> subplot(1,2,2);
>> imshow(Gato_mascara);
```





```
>> subplot(1,2,1);
>> imshow(Cachorro);
>> subplot(1,2,2);
>> imshow(Cachorro_mascara);
```

```
>> filtro = fspecial("kirsch");
>> Cachorro_mascara = imfilter(Cachorro, filtro);
>> Gato_mascara = imfilter(Gato, filtro);
>> subplot(1,2,1);
>> imshow(Gato);
>> subplot(1,2,2);
>> imshow(Gato_mascara);
>> subplot(1,2,1);
```



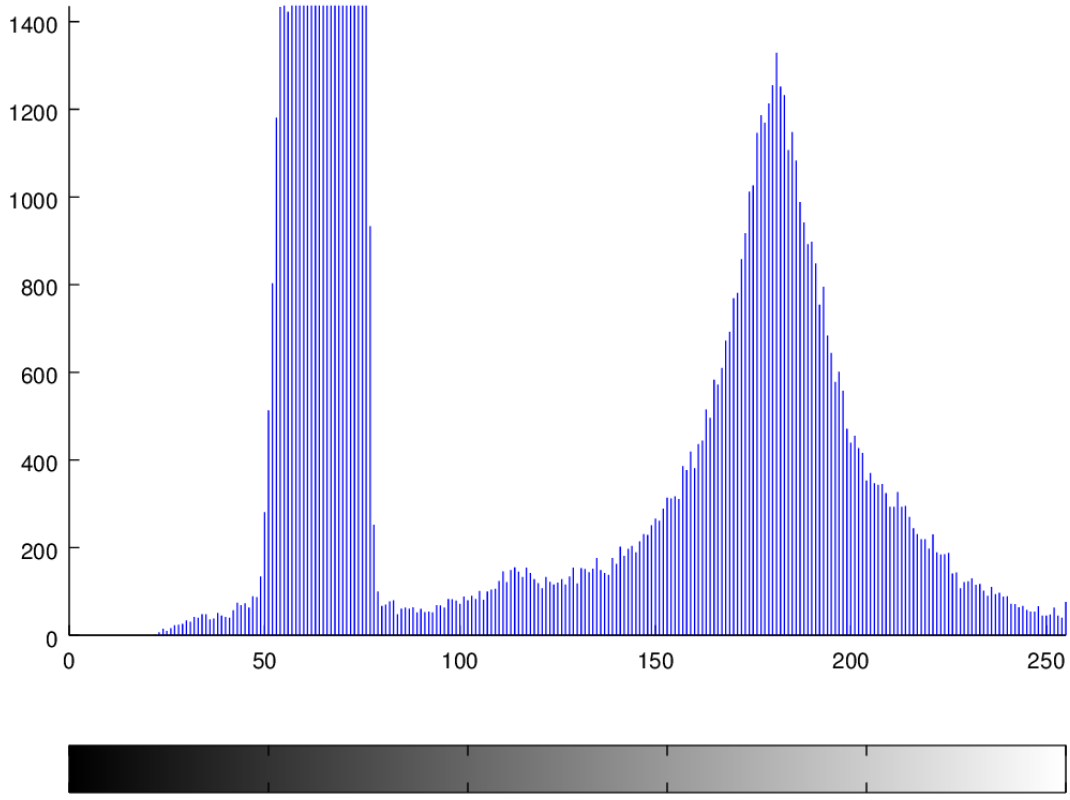
```
>> subplot(1,2,1);
>> imshow(Cachorro);
>> subplot(1,2,2);
>> imshow(Cachorro_mascara);
```

Exercício 5:

Leia a imagem Coins.png

```
>> coins = imread("Coins.png");
```

Com ajuda da função imhist identifique um limiar (de intensidade) que melhor separa as moedas do fundo da imagem (background).



Abaixo de 85 está localizado o background, então esse vai ser o limiar.

Com o limiar definido, crie uma imagem binária em que os pixels do background das moedas tenham valores diferentes.

```
>> coins2 = coins>=85;
```

Aplique esta máscara binária à imagem original de forma a mudar a cor do background na imagem resultante.

```
>> coins_mask = coins.*coins2;
```

Apresente a imagem original e a resultante em uma única figura.

```
>> subplot(1,2,1)
>> imshow(coins);
>> subplot(1,2,2)
>> imshow(coins_mask);
```