



## Relatório atividade 2

Aluno: Lyncoln Sousa de Oliveira

# Objetivos

- Manipular imagens utilizando o python
- Utilizar *Singular Value Decomposition* (SVD) para comprimir uma imagem

# Lendo a imagem

- Biblioteca matplotlib.image e matplotlib.pyplot

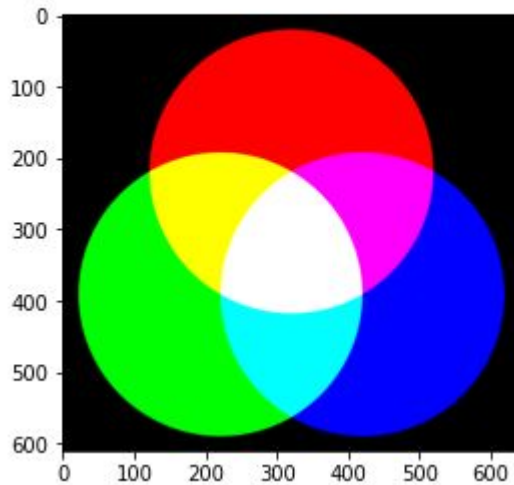
```
A = imread("teste.jpg")  
A.shape  
(610, 640, 3)
```



```
A[:3,:3,:3]  
array([[0., 0., 0.],  
       [0., 0., 0.],  
       [0., 0., 0.]],  
      [[0., 0., 0.],  
       [0., 0., 0.],  
       [0., 0., 0.]],  
      [[0., 0., 0.],  
       [0., 0., 0.],  
       [0., 0., 0.]], dtype=float32)
```

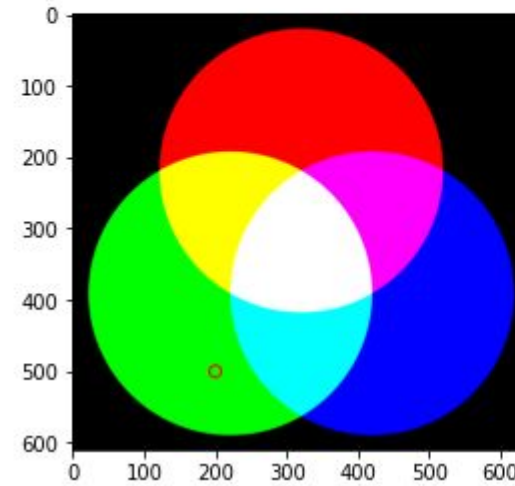
# Plotando a imagem

```
plt.imshow(A)  
plt.show()
```



```
plt.imshow(A)  
print(A[500,200,:])  
plt.scatter(x=200, y=500, facecolors='none', edgecolors='red')  
plt.show()
```

[0. 1. 0.]



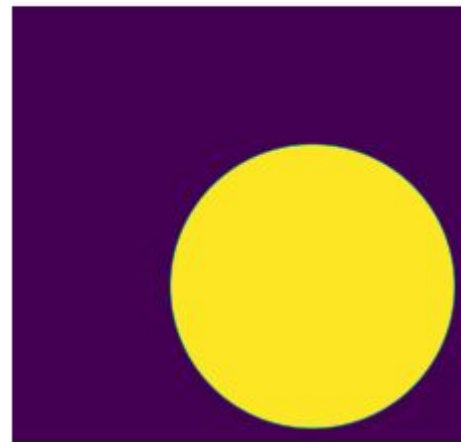
```
plt.imshow(A[:, :, 0])  
plt.axis('off')  
plt.show()
```



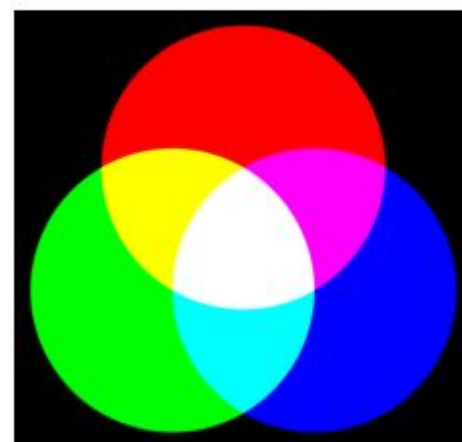
```
plt.imshow(A[:, :, 1])  
plt.axis('off')  
plt.show()
```



```
plt.imshow(A[:, :, 2])  
plt.axis('off')  
plt.show()
```



```
plt.imshow(A)  
plt.axis('off')  
plt.show()
```



# Blas/Lapack

- Função blas `scipy.linalg.blas.dgemm` para multiplicação de matrizes

`scipy.linalg.blas.dgemm`

```
scipy.linalg.blas.dgemm(alpha, a, b[, beta, c, trans_a, trans_b, overwrite_c]) = <fortran object>
```

Wrapper for `dgemm`.

Parameters:     **alpha** : input float  
                  **a** : input rank-2 array('d') with bounds (lda,ka)  
                  **b** : input rank-2 array('d') with bounds (ldb,kb)  
  
Returns:         **c** : rank-2 array('d') with bounds (m,n)

- Função `scipy.linalg.lapack.dgesdd` para realizar svd

`scipy.linalg.lapack.dgesdd`

```
scipy.linalg.lapack.dgesdd(a[, compute_uv, full_matrices, lwork, overwrite_a]) = <fortran object>
```

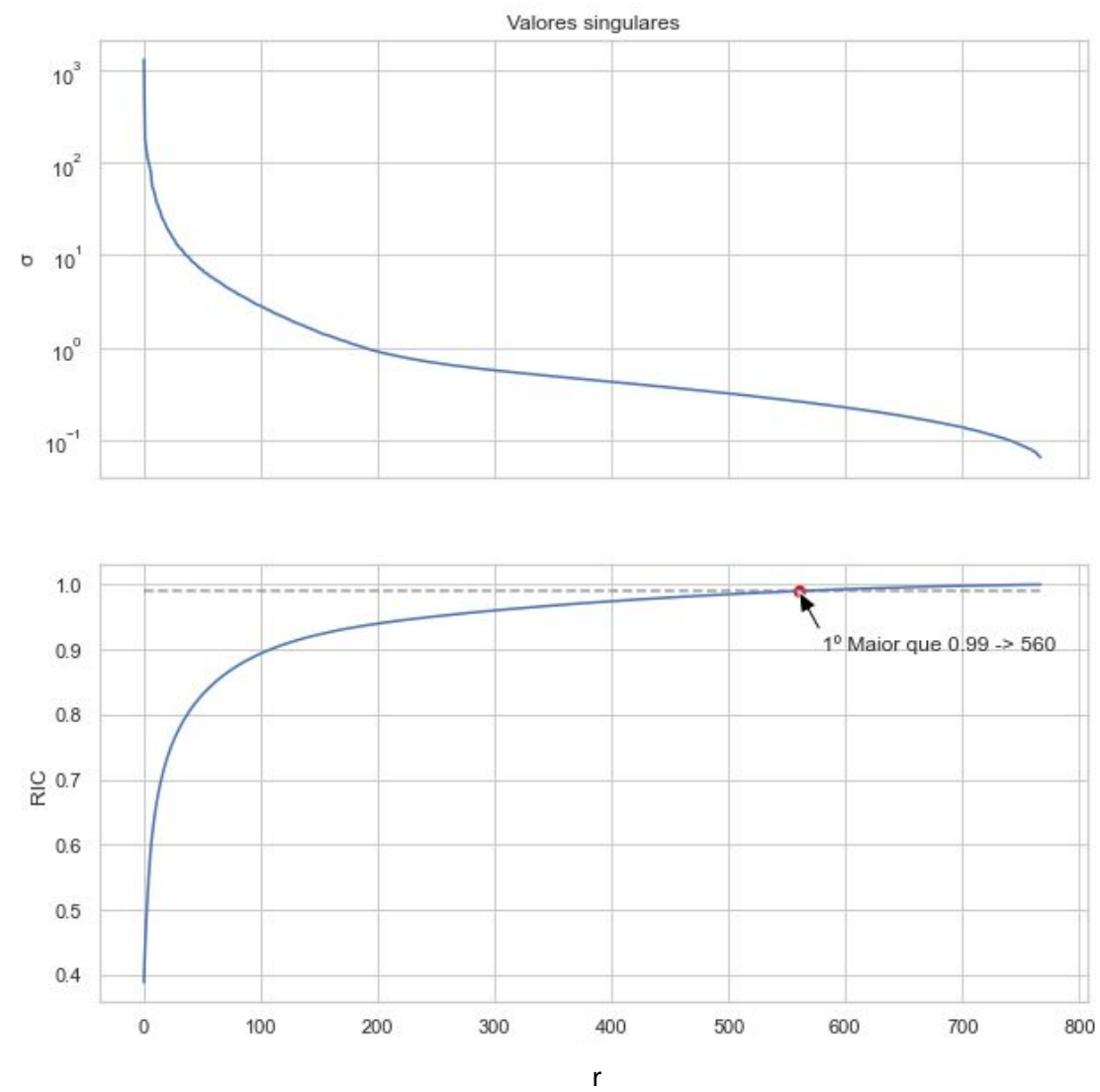
Wrapper for `dgesdd`.

Parameters:   **a** : input rank-2 array('d') with bounds (m,n)  
  
Returns:       **u** : rank-2 array('d') with bounds (u0,u1)  
              **s** : rank-1 array('d') with bounds (minmn)  
              **vt** : rank-2 array('d') with bounds (vt0,vt1)  
              **info** : int

# Utilizando o SVD

- Realizado a decomposição para cada matriz utilizando  $r$  valores singulares
- Agrupando os resultados finais em uma única matriz para plotar
- Imagem original de tamanho 1024x768







# Resultados

$r = 5$ , RIC = 0.57



$r = 20$ , RIC = 0.74



$r = 140$ , RIC = 0.92



$r = 280$ , RIC = 0.96



$r = 560$ , RIC = 0.99



Original



$r = 280$ , RIC = 0.96



# Conclusão

- Álgebra linear em todo lugar
- Ótima maneira de representar a imagem comprimida



fonte: <https://timbaumann.info/svd-image-compression-demo/>

Em relação aos números alocados na matriz:

Tamanho de  $A$ :  $1024 \times 768 \times 3 = 2.359.296$

Representação comprimida:  $(k \times 768 + k + k \times 1024) \times 3, \{k=280\} = 1.506.120$

Redução de 36%

- Pode piorar dependendo do valor de  $k$