

Frobenius

October 7, 2020

```
[1]: # Python code: Frobenius eigenvectors
```

```
import numpy as np

def Frobenius(matrix):
    eval, evec = np.linalg.eig(matrix)
    dominant = max(abs(eval)).astype(np.complex)
    frob = evec[:,eval == dominant].real
    frob = frob/frob.sum()
    return(frob)
```

```
[2]: # Some tests that you can run
```

```
A = np.array([[0,0,1/3,1/2],[1/2,0,1/3,1/2],[1/2,1,0,0],[0,0,1/3,0]])
B = np.array([[0,3,0,0,1,2],[3,0,2,2,2,1],[6,4,0,2,1,1],[3,1,1,0,2,2],
              [2,1,2,4,0,2],[1,2,2,4,4,0]])
```

```
[3]: Frobenius(A)
```

```
[3]: array([[0.19354839],
            [0.29032258],
            [0.38709677],
            [0.12903226]])
```

```
[4]: Frobenius(B)
```

```
[4]: array([[0.10801061],
            [0.15830238],
            [0.19691403],
            [0.14635945],
            [0.17822515],
            [0.21218838]])
```

```
[ ]:
```