

# Project6

May 31, 2020

```
[21]: import numpy as np

#Project 3 Matrix
m1 = np.matrix('0.38 0.73 ; 0.62 0.27')
w1, v1 = np.linalg.eig(m1)
print("Project 3 Matrix:")
print(m1)
print("Results:")
print(w1)
print(v1)
print("Eigenvalue 1 & Eigenvector 1:")
print(w1[0])
print(v1[:,0])
print("Eigenvalue 2 & Eigenvector 2:")
print(w1[1])
print(v1[:,1])
```

Project 3 Matrix:

```
[[0.38 0.73]
 [0.62 0.27]]
```

Results:

```
[ 1.   -0.35]
[[ 0.7621968742 -0.7071067812]
 [ 0.6473452904  0.7071067812]]
```

Eigenvalue 1 & Eigenvector 1:

```
1.0
[[0.7621968742]
 [0.6473452904]]
```

Eigenvalue 2 & Eigenvector 2:

```
-0.35
[[-0.7071067812]
 [ 0.7071067812]]
```

The results of getting the eigenvalues and eigenvectors for the matrix in project 3 are above. I read that a steady-state vector of a regular Markov chain is an eigenvector for the transition matrix corresponding to the eigenvalue 1, so that would be the 1st Eigenvalue and Eigenvector. The 2nd eigenvalue and eigenvector doesn't really make sense because they have negative numbers. The eigenvector corresponding to 1 holds the values 0.7621968742 and 0.6473452904, which represent

the steady-state vector, meaning the approximate final vector reached if the Markov chain is run infinite number of times. The two numbers also directly correlate to the project 3 results of the vector [54.07396169, 45.92603831], which was reached after running the Markov chain 10 times. The ratio of the eigenvector values and the project 3 vector results show that the higher value is  $\sim 1.1774$  times larger than the lower value. This shows that the eigenvector represents the steady-state vector of the Markov chain. So the Eigenvector shows that if the total percentage is 141 and the vector values are normalized to represent a total percentage of 100%, then the total number of cars that end up in city A is 54.074% and the percent of cars that end up in city B is 45.926%.

```
[23]: #Project 5 Matrix
m2 = np.matrix('1.0000011074 0 0 0; 0.0000655823 0.9914979223 0 0; 0 0.
↪0.0069673501 1 0; 0 0.0015347277 0 1')
w2, v2 = np.linalg.eig(m2)
np.set_printoptions(suppress=True)
print("Project 5 Matrix:")
print(m2)
print("Results:")
print(w2)
print(v2)
print("Eigenvalue 1 & Eigenvector 1:")
print(w2[0])
print(v2[:,0])
print("Eigenvalue 2 & Eigenvector 2:")
print(w2[1])
print(v2[:,1])
print("Eigenvalue 3 & Eigenvector 3:")
print(w2[2])
print(v2[:,2])
print("Eigenvalue 4 & Eigenvector 4:")
print(w2[3])
print(v2[:,3])
```

Project 5 Matrix:

```
[[1.0000011074 0.          0.          0.          ]
 [0.0000655823 0.9914979223 0.          0.          ]
 [0.          0.0069673501 1.          0.          ]
 [0.          0.0015347277 0.          1.          ]]
```

Results:

```
[1.          1.          0.9914979223 1.0000011074]
[[ 0.          0.          0.          0.0201212716]
 [ 0.          0.          0.7660316554 0.0001551888]
 [ 0.          1.         -0.6277536996 0.9763905092]
 [ 1.          0.         -0.1382779648 0.2150736706]]
```

Eigenvalue 1 & Eigenvector 1:

```
1.0
[[0.]
 [0.]
 [0.]
```

```

[1.]]
Eigenvalue 2 & Eigenvector 2:
1.0
[[0.]
 [0.]
 [1.]
 [0.]]
Eigenvalue 3 & Eigenvector 3:
0.9914979223
[[ 0.          ]
 [ 0.7660316554]
 [-0.6277536996]
 [-0.1382779648]]
Eigenvalue 4 & Eigenvector 4:
1.0000011074
[[0.0201212716]
 [0.0001551888]
 [0.9763905092]
 [0.2150736706]]

```

The function `numpy.linalg.eig()` returns 4 eigenvalues and their respective eigenvectors for the matrix from project 5. The daily vector contains the numbers representing [susceptible, infected, recovered, and died]. If the eigenvectors represent the steady-state vectors corresponding to each eigenvalue, then the four eigenvectors can represent the following: 1. The first eigenvector shows a steady state where it looks like everyone dies. 2. The second eigenvector shows a steady state where everyone recovers. 3. The third eigenvector shows the most people ending up infected and a negative value for recovered and deaths, which does not make sense. 4. The fourth vector shows that over time, the amount of susceptible people has a value of 0.02, or 1.66% of the total amount of people. The amount of people infected has a value of 0.000155, and represents 0.0013% of the population. The amount of people recovered has a value of 0.9764, and represents 80.58% of the population. The number of deaths has a value of 0.215 and represents 17.75% of the population. The fourth eigenvector shows the most believable model out of the four. It shows that over time everyone will eventually get sick and that over an infinite amount of time about 2% of the population will be susceptible to the covid and 0.0013% of the population will be infected by the disease. If all people lived long enough to contract the virus and recover, then 80.58% of the population would get the disease and survive, however 17.75% of the population would eventually die.

[ ]: