Name: Linh Tang

Date: 08/05/2020

Math bootcamp

PROJECT 1 – MANGROVE

Problem 1:

In this project, matrix models were used to simulate population change of long-lived species – mangroves in Southeastern Australia. Studying the matrix model could predict the long- time behavior, population proportion and some characteristics of the population after a regeneration. The regeneration occurs at two scale: small scale - type a (due to the death of a few individuals), large scale - type b (the disturbance comes from storms, pathogen attack).

The life stage of mangrove consists of seven stages: propagules, cotyledonary seedlings, seedlings, saplings, young tree, tree, older tree. The propagules grow on older tree, then they fall from the tree and disperse into the surroundings; they commonly land in water and continue their embryonic development. Cotyledonary seedlings are the primary leaf of the embryo of seed plants. Seedlings is the stage occurs after the post- cotyledonary phase. Established seedlings recruit to the following stage – saplings. When the saplings grow, it develops its roots structure. The last three stages of mangrove's life cycle are young tree, tree, then older tree.

Problem 2:

The following matrix is the matrix model of regeneration after a “type a” disturbance; this is built by using the data from Table 1.

```
M = np.zeros((7,7))
M[0,5] = 500
M[0,6] = 1000
M[1,0] = 0.2
M[1,1] = 0.666
M[2,1] = 0.083
M[2,2] = 0.825
M[3,2] = 0.01
M[3,3] = 0.909
M[4,3] = 0.073
M[4,4] = 0.963
M[5,4] = 0.008
M[5,5] = 0.98
M[6,5] = 0.012
M[6,6] = 0.999
print (M)
```

```
[[0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 5.00e+02 1.00e+03]
 [2.00e-01 6.66e-01 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00]
 [0.00e+00 8.30e-02 8.25e-01 0.00e+00 0.00e+00 0.00e+00 0.00e+00]
 [0.00e+00 0.00e+00 1.00e-02 9.09e-01 0.00e+00 0.00e+00 0.00e+00]
 [0.00e+00 0.00e+00 0.00e+00 7.30e-02 9.63e-01 0.00e+00 0.00e+00]
 [0.00e+00 0.00e+00 0.00e+00 0.00e+00 8.00e-03 9.80e-01 0.00e+00]
 [0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 1.20e-02 9.99e-01]]
```

Problem 3:

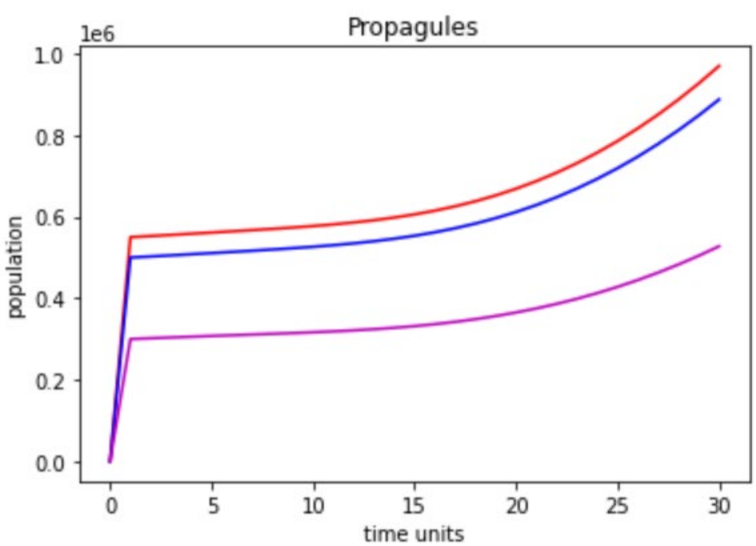
The model was simulated in three different initial conditions and run in 30 time units:

A_1= [100,500,800,600,500,500,300]

A_2 = [150,1200,200,600,400,600,200]

A_3 = [900,400,300,100,500,100,250]

```
#Problem 3
#initial conditions
A_1= np.array(([100,500,800,600,500,500,300]))
A_2 = np.array(([150,1200,200,600,400,600,200]))
A_3 = np.array(([900,400,300,100,500,100,250]))

t_units = 30 #30 time units

t1 = A_1
for i in range (t_units):
  A_1 = np.matmul(M,A_1)
  t1 = np.vstack((t1,A_1)) #adding row by row

pri#nt ("test" , t1)

t2 = A_2
for i in range (t_units):
  A_2 = np.matmul(M,A_2)
  t2 = np.vstack ((t2,A_2))

t3 = A_3
for i in range (t_units):
  A_3 = np.matmul(M,A_3)
  t3 = np.vstack ((t3,A_3))

#plot

x = np.arange(0,t_units + 1,1)
y1 = t1[:,0] #extract first column for propagules
y2 = t2[:,0]
y3 = t3[:,0]
#print (y1)

plt.plot(x,y1,'r',x,y2,'b', x,y3, 'm')
plt.title ("Propagules")
plt.xlabel('time units')
plt.ylabel('population')
```



Problem 4:

There are total seven eigen values with their corresponding eigen vectors. The dominant eigen vector of this model is 1.048 which is greater than 1. Therefore, the mangrove population will grow in the long run.

```
[141] #Problem 4
      v, w = np.linalg.eig(M)
      print("Eigenvalue\n",v)

      print("Eigenvector\n",w)


      max_eigval = np.max(abs(v));

      print ("Dominant eigen value is: ", max_eigval)
      if max_eigval > 1:
        print ("In the long run, the mangrove population will grow")
      else:
        print ("In the long run, the mangrove population will decay")
```

```
Eigenvalue
 [-1.00306923e-04+0.j          6.97792977e-01+0.j
  7.35730360e-01+0.j          9.42674610e-01+0.09655739j
  9.42674610e-01-0.09655739j  1.04814018e+00+0.j
  9.75087570e-01+0.j        ]
Eigenvector
 [[ 9.57358483e-01+0.00000000e+00j -1.31921008e-01+0.00000000e+00j
   2.47202261e-01+0.00000000e+00j -7.89004161e-01+0.00000000e+00j
  -7.89004161e-01-0.00000000e+00j -8.72986671e-01+0.00000000e+00j
   7.05065522e-01+0.00000000e+00j]
 [-2.87451746e-01+0.00000000e+00j -8.29875138e-01+0.00000000e+00j
   7.09023329e-01+0.00000000e+00j -5.08424143e-01+1.77436253e-01j
  -5.08424143e-01-1.77436253e-01j -4.56893423e-01+0.00000000e+00j
   4.56223795e-01+0.00000000e+00j]
 [ 2.89158721e-02+0.00000000e+00j  5.41476681e-01+0.00000000e+00j
  -6.59226771e-01+0.00000000e+00j -1.52941554e-01+2.50647489e-01j
  -1.52941554e-01-2.50647489e-01j -1.69947673e-01+0.00000000e+00j
   2.52296542e-01+0.00000000e+00j]
 [-3.18071305e-04+0.00000000e+00j -2.56372480e-02+0.00000000e+00j
   3.80462944e-02+0.00000000e+00j  1.82184736e-02+2.21931810e-02j
   1.82184736e-02-2.21931810e-02j -1.22141336e-02+0.00000000e+00j
   3.81760957e-02+0.00000000e+00j]
 [ 2.41088131e-05+0.00000000e+00j  7.05682333e-03+0.00000000e+00j
  -1.22206358e-02+0.00000000e+00j  1.32903787e-02-1.65712927e-02j
   1.32903787e-02+1.65712927e-02j -1.04725145e-02+0.00000000e+00j
   2.30555428e-01+0.00000000e+00j]
 [-1.96786495e-07+0.00000000e+00j -2.00046711e-04+0.00000000e+00j
   4.00234292e-04+0.00000000e+00j -1.56479944e-03-4.96246669e-04j
  -1.56479944e-03+4.96246669e-04j -1.22952593e-03+0.00000000e+00j
  -3.75464609e-01+0.00000000e+00j]
 [ 2.36356443e-09+0.00000000e+00j  7.96980266e-06+0.00000000e+00j
  -1.82429372e-05+0.00000000e+00j  3.86255276e-05+1.71939156e-04j
   3.86255276e-05-1.71939156e-04j -3.00249441e-04+0.00000000e+00j
   1.88419805e-01+0.00000000e+00j]]
Dominant eigen value is:  1.048140178442451
In the long run, the mangrove population will grow
```

Problem 5:

If we had 1000 older trees, we would have 2907538 propagules, 1521712 Cotyledonary, 566021 seedlings, 40679 saplings, 34879 young trees, 4095 trees, 1000 older trees.

```
[142] #Problem 5

print ("Dominant eigen value is: ", max_eigval)

eig_vect =[]
for i in range (len(v)):
  if v[i] == max_eigval:
    eig_vect = w[:,i]
    print ("Dominant eigen vector is: \n", eig_vect)

new_vect = (eig_vect/eig_vect[6])*1000

print ("New population when we have 1000 older trees:\n",new_vect)
```

```
Dominant eigen value is:  1.048140178442451
Dominant eigen vector is: 
 [-8.72986671e-01+0.j -4.56893423e-01+0.j -1.69947673e-01+0.j
 -1.22141336e-02+0.j -1.04725145e-02+0.j -1.22952593e-03+0.j
 -3.00249441e-04+0.j]
New population when we have 1000 older trees:
 [2.90753804e+06+0.j 1.52171282e+06+0.j 5.66021613e+05+0.j
 4.06799545e+04+0.j 3.48793805e+04+0.j 4.09501487e+03+0.j
 1.00000000e+03+0.j]
```

Problem 6:

If we want to promote mangrove regeneration after a type a disturbance, we should focus on changing parameter “tree” (row 5, column 5 of the matrix model). The effectiveness was determined by comparing the dominant eigen value of each matrix with the original dominant eigen value. Fourteen matrixes were built by changing each parameter of the original matrix M once at a time.

```
M_list = np.array([M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,M11,M12,M13,M14])
eigv_list =[]
for i in range (len(M_list)):
  v1, w1 = np.linalg.eig(M_list[i])
  max_v = np.max(abs(v1))
  eigv_list.append(max_v)

#print (eigv_list)
diff = [] #difference with the original dominant eig val
for k in range (len(eigv_list)):
  d = eigv_list[k] - max_eigval
  diff.append(d)


eff = np.max(diff)



#return the position
for n in range (len(diff)):
  if diff[n] == eff:
    print("The most effective change belongs to M",n+1)

print ("The change of parmeter at row 5, col 5")
```

```
The most effective change belongs to M 12
The change of parmeter at row 5, col 5
```

Problem 7:

After building the matrix model for type b, we got the dominant eigen value of this one was 1.222. Similar to type a, the mangrove population for type b also grow in the long run.

```
#Problem 7
N = np.zeros((7,7))

N[0,4] = 100
N[0,5] = 500
N[0,6] = 1000
N[1,0] = 0.4
N[1,1] = 0.666
N[2,1] = 0.23
N[2,2] = 0.825
N[3,2] = 0.045
N[3,3] = 0.909
N[4,3] = 0.045
N[4,4] = 0.963
N[5,4] = 0.008
N[5,5] = 0.98
N[6,5] = 0.012
N[6,6] = 0.999

print ("Matrix for type b\n",N,"\n")

v, w = np.linalg.eig(N)
print("Eigenvalue\n",v, "\n")

print("Eigenvector\n",w,"\n")

max_val = np.max(abs(v));

print ("Dominant eigen value is: ", max_val,"\n")
if max_val > 1:
  print ("In the long run, the mangrove population type b will grow")
else:
  print ("In the long run, the mangrove population type b  will decay")
```

```
Matrix for type b
 [[0.00e+00 0.00e+00 0.00e+00 0.00e+00 1.00e+02 5.00e+02 1.00e+03]
 [4.00e-01 6.66e-01 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00]
 [0.00e+00 2.30e-01 8.25e-01 0.00e+00 0.00e+00 0.00e+00 0.00e+00]
 [0.00e+00 0.00e+00 4.50e-02 9.09e-01 0.00e+00 0.00e+00 0.00e+00]
 [0.00e+00 0.00e+00 0.00e+00 4.50e-02 9.63e-01 0.00e+00 0.00e+00]
 [0.00e+00 0.00e+00 0.00e+00 0.00e+00 8.00e-03 9.80e-01 0.00e+00]
 [0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 1.20e-02 9.99e-01]]

Eigenvalue
 [0.04692915+0.j         0.35496315+0.j         0.88923912+0.35210781j
 0.88923912-0.35210781j 1.22261572+0.j         0.96950687+0.00943716j
 0.96950687-0.00943716j]

Eigenvector
 [[ 8.29293670e-01+0.00000000e+00j  5.72366531e-01+0.00000000e+00j
    6.58391202e-01+0.00000000e+00j  6.58391202e-01-0.00000000e+00j
   -7.68574035e-01+0.00000000e+00j  4.48238740e-02+1.21803423e-01j
    4.48238740e-02-1.21803423e-01j]
 [-5.35831186e-01+0.00000000e+00j -7.36075525e-01+0.00000000e+00j
    3.38240429e-01-5.33495642e-01j  3.38240429e-01+5.33495642e-01j
   -5.52319319e-01+0.00000000e+00j  6.40041466e-02+1.58537932e-01j
    6.40041466e-02-1.58537932e-01j]
 [ 1.58393253e-01+0.00000000e+00j  3.60178932e-01+0.00000000e+00j
   -2.98248033e-01-2.75354554e-01j -2.98248033e-01+2.75354554e-01j
   -3.19487982e-01+0.00000000e+00j  1.17846469e-01+2.44636036e-01j
    1.17846469e-01-2.44636036e-01j]
 [-8.26810973e-03+0.00000000e+00j -2.92544656e-02+0.00000000e+00j
   -3.29478552e-02+3.99657136e-02j -3.29478552e-02-3.99657136e-02j
   -4.58425978e-02+0.00000000e+00j  1.13266028e-01+1.64274098e-01j
    1.13266028e-01-1.64274098e-01j]
 [ 4.06153015e-04+0.00000000e+00j  2.16508416e-03+0.00000000e+00j
    5.73797942e-03+3.00878036e-03j  5.73797942e-03-3.00878036e-03j
   -7.94604009e-03+0.00000000e+00j  7.83321533e-01+0.00000000e+00j
    7.83321533e-01-0.00000000e+00j]
 [-3.48229089e-06+0.00000000e+00j -2.77114434e-05+0.00000000e+00j
    3.25906162e-05-1.38769395e-04j  3.25906162e-05+1.38769395e-04j
   -2.62012376e-04+0.00000000e+00j -3.30156777e-01-2.96931790e-01j
   -3.30156777e-01+2.96931790e-01j]
 [ 4.38911565e-08+0.00000000e+00j  5.16332754e-07+0.00000000e+00j
   -4.62603655e-06+3.31348634e-07j -4.62603655e-06-3.31348634e-07j
   -1.40604987e-05+0.00000000e+00j  8.67885428e-02+1.48584407e-01j
    8.67885428e-02-1.48584407e-01j]]

Dominant eigen value is:  1.2226157176665158

In the long run, the mangrove population type b will grow
```

Problem 8:

Question: Calculating how many individuals of each stage when we have 1000 older trees for type b, then compare the data with the result we got in type a (Problem 5). Writing codes to determine the rate of change of each model.

In type b model, when we have 1000 older trees, there would be 54661933 propagules, 39281631 contyledonary seedlings, 22722379 seedlings, 3260382 saplings, 565132 young trees, 18635 trees, 1000 older trees. Compared with what we got in problem 5 for type a, the number of individuals of other six stages in type b are a lot greater than the number in type a. Therefore, we can see that the population growth rate from the disturbance type a was less than the one from type b.

In the long run, the long- term behavior of type a and type b both are exponential growth. The difference is that type b changes at the rate 22%, while type a changes at the rate 5%.

```python
#Problem 8

print ("Dominant eigen value is: ", max_eval)

e_vect =[]
for i in range (len(v_b)):
  if v_b[i] == max_eval:
    e_vect = w_b[:,i]
    print ("Dominant eigen vector is: \n", e_vect)

new = (e_vect/e_vect[6])*1000

print ("\nWhen we have 1000 older trees for tybe b model, there will be: ",new.round(), "\n\n")

# Analyzing the behavior and rate of change
def rate (val):
  if abs(val) > 1:
    p = (abs(val) -1)*100
    if val > 0:
      print ("Long term behavior: Exponential growth at the rate ",p.round(), "percent.")
    else:
      print ("Long term behavior: Fluctuating growth at the rate ",p.round(), "percent.")
  elif abs (val) < 1:
    q = (1 - abs(val))*100
    if val > 0:
      print ("Long term behavior: Exponential decay at the rate ",q.round(), "percent.")
    elif val < 0:
      print ("Long term behavior: Fluctuating decay at the rate ",q.round(), "percent.")

print ("-Model of type a:")
rate (max_eigval)

print ("-Model of type b:")
rate (max_eval)
```

```
Dominant eigen value is:  1.2226157176665158
Dominant eigen vector is: 
 [-7.68574035e-01+0.j -5.52319319e-01+0.j -3.19487982e-01+0.j
 -4.58425978e-02+0.j -7.94604009e-03+0.j -2.62012376e-04+0.j
 -1.40604987e-05+0.j]

When we have 1000 older trees for tybe b model, there will be:  [5.4661933e+07+0.j 3.9281631e+07+0.j 2.2722379e+07+0.j 3.2603820e+06+0.j
 5.6513200e+05+0.j 1.8635000e+04+0.j 1.0000000e+03+0.j]


-Model of type a:
Long term behavior: Exponential growth at the rate  5.0 percent.
-Model of type b:
Long term behavior: Exponential growth at the rate  22.0 percent.
```