```
#-------------------------------------------------------------------------------
# Name:        Markov Chain Land Cover Change Model
# Purpose:     Simple explanation of model with two land cover states
#
# Author:      Eugenio Arima, UT Austin
#
# Created:     07/03/2017
# Copyright:   (c) ea9267 2017
# Licence:     <your licence>
# Modifications:
#
#-------------------------------------------------------------------------------
# Import system modules
import sys, string, os
import numpy as np
import matplotlib.pyplot as plt
#suppose you have only two states, forest and deforested
# the transition matrix probability is
tmp = np.array([0.8, 0.2, 0.1, 0.9]).reshape(2,2)
print(tmp)
# probability of forest --> remain forest = 0.8
# probability of forest --> deforest = 0.2
# probability of deforest --> regrowth back to forest = 0.1
# probability of deforest --> remain deforest = 0.9
# Suppose initial state is forest
lc0 = np.array([[1.0,0]])
#python: the double square brackets forces the array to be 2D instead of
# a 1D array (important later on when we concatenate arrays)

# What is the probability of forest or deforested after 3 periods?

# Can calculate in two ways.
# 1) As a Markov Chain
# p(t) = f(p(t-1))
lc1 = np.dot(lc0, tmp)
lc2 = np.dot(lc1, tmp)
lc3 = np.dot(lc2, tmp)
print(lc3)

# 2) As matrix exponentiation
tmp2 = np.dot(np.dot(tmp, tmp), tmp)
lc3_2 = np.dot(lc0,tmp2)
print(lc3_2)

#Both return the same result after three periods
# Probability of forest = 0.562
# probability of deforested = 0.438

tmp_0 = tmp
# Convergence, steady state
# run the same MC for a long time
#Let's also store the probabilites in an array lcprob
lcprob = lc0
for t in range(1,30):
```

```
        tmp_f = np.dot(tmp_0, tmp)
        tmprob = np.dot(lc0, tmp_f)
        lcprob = np.concatenate((lcprob, tmprob),axis=0)
        tmp_0 = tmp_f

    # Steady state probability
    lc_f = np.dot(lc0, tmp_f)
    print("Probability remain forested: {:.4f}".format(lc_f[0][0]))
    print("Probability deforested: {:.4f}".format(lc_f[0][1]))
    # 1/3 chance it will be forest
    # 2/3 chance it will deforested

    #plot steady state convergence
    t = np.arange(1, 31)

    plt.plot(t, lcprob[:,0], 'g:', label = "Forested", marker = 'o')
    plt.plot(t, lcprob[:,1], 'b-', label = "Deforested", marker = 'x')
    plt.xlabel('Time')
    plt.ylabel('Probability')
    plt.xlim(1, 30)
    plt.ylim(0, 1)
    plt.legend(loc=1, prop={'size':11})
    plt.title('Steady State Land Cover Probabilities')
    plt.show()
```