확률통계론 2018125063 조성완

import sys

import math

import numpy as np

import matplotlib.pyplot as plt

def random\_variable(step): => 함수 정의

arr = np.zeros((4,4)) => 4X4 배열 생성하고 0으로 초기화

px, py = np.random.choice(4,step, p=[0.2, 0.4, 0.3, 0.1]), np.random.choice(4,step, p=[0.25, 0.25, 0.25, 0.25])

* random X,Y를 step만큼 주어진 확률대로 뽑기
* 0~3까지의 숫자가 주어진 확률에 의해 step만큼 배열로 들어감

pX\_exp, pY\_exp, pX\_geo, pY\_geo = [], [], [], [] => E[X],E[Y],E[X^2],E[Y^2] 변수 초기화

px4, py1, pxy23 = 0, 0, 0 => 문제 3 출력할 변수 초기화

EX1, EY1, EX3, EY3 = 0, 0, 0, 0 => E[X],E[Y],E[X^2],E[Y^2] 변수 초기화

for i in range(step): => arr 배열에 위에서 랜덤으로 뽑은 (x,y)에 더해주기

arr[px[i], py[i]] += 1

for i in range(4): =>실험값 바탕으로 arr의 x,y의 개수를 pX\_exp, pY\_exp 에 넣어줌

temp1, temp2 = 0, 0

for j in range(4): => (i,j)의 개수 더하기 <= x축 기준

temp1 += arr[i,j]

temp2 += arr[j,i]

pX\_exp.append(temp1) => 한 줄 다 더한 거 pX에 넣어주기

pY\_exp.append(temp2)

for i in range(4): => numpy를 쓰기 위해서 pX\_exp, pY\_exp를 변환

j, k = int(pX\_exp[i]), int(pY\_exp[i])

for a in range(j): => i+1 을 pX\_geo에 pX\_exp[i] 개 넣어줌

pX\_geo.append(i+1)

for b in range(k):

pY\_geo.append(i+1)

for i in range(4): => X=i인 (배열에서는 i-1) 모든 성분을 더해서 확률 구함

px4 += arr[3,i]/step

py1 += arr[i,0]/step

pxy23 += arr[i,2] => 조건부에서 y=3 인 성분들 중에 x=2인 성분들의 확률 구함

pxy23 = arr[1,2]/ pxy23

for i in range(4): => X,Y 의 평균구하기

EX1 += (i+1) \* pX\_exp[i] / step

EY1 += (i+1) \* pY\_exp[i] / step

EX2, EY2 = np.mean(pX\_geo), np.mean(pY\_geo) => np.mean을 활용한 평균 구하기

for i in range(4): => X^2 평균 구하기

EX3 += pow((i+1), 2) \* pX\_exp[i] / step

EY3 += pow((i+1), 2) \* pY\_exp[i] / step

varX, varY = (EX3 - pow(EX1,2)), (EY3 - pow(EY1,2)) => var 식을 활용해 분산 구하기

varX\_var, varY\_var = np.var(pX\_geo), np.var(pY\_geo) => 문제 5 np.var 을 활용해 분산 구하기

pxy00 = arr[0,0] / step => Pxy(x,y) == Px(x) \* Py(y) 판정 하기 위한 변수 선언

px0 = (arr[0,0] + arr[0,1] + arr[0,2] + arr[0,3]) / step

py0 = (arr[0,0] + arr[1,0] + arr[2,0] + arr[3,0]) / step

print(' Experiment Geometric')

print('X\_Mean: %.4f, %.4f' %(EX1, EX2))

print('Y\_Mean: %.4f, %.4f' %(EY1, EY2))

print('X\_Variance: %.4f, %.4f' %(varX, varX\_var))

print('Y\_Variance: %.4f, %.4f' %(varY, varY\_var))

print('\nPx(4) = %.4f\nPy(1) = %.4f\nPxy(2|3) = %.4f' %(px4,py1,pxy23)) => 문제 3 출력

print("\nE[2X+4] = %.4f\nE[-Y^2-1] = %.4f\nvar[3Y-3] = %.4f" %(((2 \* EX1) + 4), (-EY3 -1), (9 \* varY))) => 문제 6 출력

if (pxy00 == px0\*py0): => pxy00 == px0\*py0 조건을 만족하면 독립, 아니면 종속 출력

print("\nindependent")

else:

print("\ndependent")

plt.figure(figsize=(4,4)) => 그래프 그리기

for x in range (4):

for y in range (4):

plt.scatter(x+1, y+1, color='b')

plt.text(x+1.1,y+1.1, "({}/{})".format(int(arr[x,y]),step), fontsize=7)

plt.show()

def main(): => 시스템에서 입력받기

if len(sys.argv) != 2:

print('usage: ./geo.py head\_prob step')

sys.exit(1)

step = int(sys.argv[1])

random\_variable(step)

if \_\_name\_\_ == '\_\_main\_\_':

main()

