



Hw 5

결과와 코드 작성 과정



과목명: 시뮬레이션 기초 및 실습

교수: 김지범

학과: 컴퓨터공학부

학번: 202201479

이름: 박지원

제출일: 2023년 05월 26일

목차

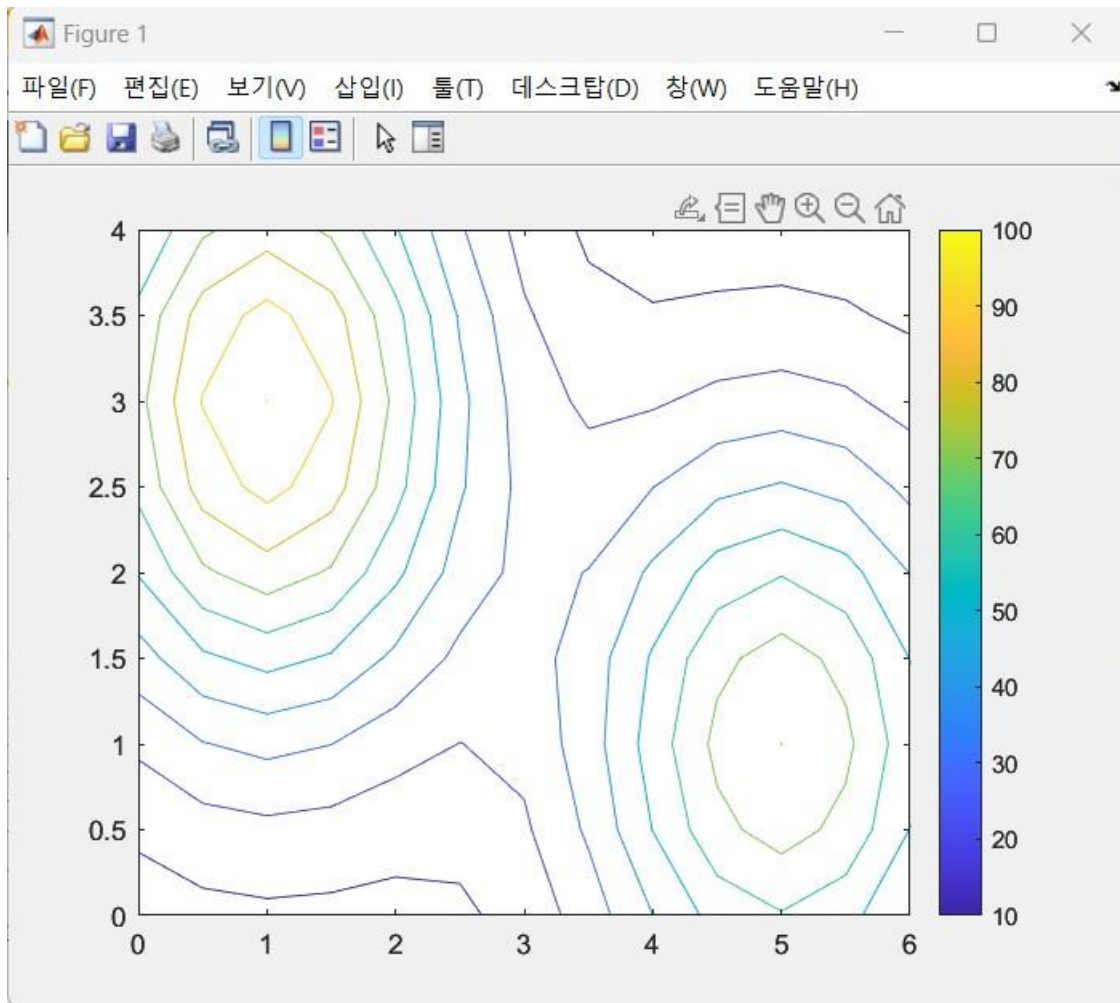
I. 문제 1번

- i. 결과 스크린 샷
- ii. 코드 작성 과정

II. 문제 2번

- i. 1) transition matrix
- ii. 2) 결과 스크린 샷
- iii. 2) 코드 작성 과정
- iv. 3) 결과 스크린 샷
- v. 3) 코드 작성 과정

I. 문제 1번 _ i. 결과 스크린 샷



I. 문제 1번 _ ii. 코드 작성 과정

fOnGrid 대신 meshgrid 를 사용한 코드 작성 과정에 대해서술하겠습니다. fOnGrid 를 이용했을 때와 같은 지 비교해야 하므로, X 와 y 의 범위는 수정하지 않았습니다. 그리고 $[x,y] = \text{meshgrid}(x,y);$ 라고 작성하여 2 차원 그리드 좌표를 반환했습니다. 그리고 나머지 코드는 그대로 작성하였습니다. 따라서 fOnGrid 를

사용하였을 때와 meshgrid 를 사용하였을 때의 결과가 같게 나옴을 알 수 있습니다.

II. 문제 2 번 _ i . 1) transition matrix

Transition matrix의 각 열의 합이 1인 이유는, 하나의 열은 어떤 상태에 대한 모든 전이 확률을 나타내기 때문입니다. 즉, 하나의 상태에 대한 모든 전이 확률을 나타내는 것이기 때문에 합은 1이 되어야 합니다.

II. 문제 2번 _ ii. 2) 결과 스크린 샷

```
>> transition_matrix
n: 4

n =

     4

P =

    0.3163    0.0827    0.3598    0.0383
    0.1368    0.1009    0.3228    0.0465
    0.2464    0.4336    0.2967    0.4196
    0.3005    0.3828    0.0207    0.4956

ans =

    1.0000    1.0000    1.0000    1.0000
```

```
>> transition_matrix
n: 10

n =

    10

P =

    0.1448    0.1309    0.1862    0.1522    0.0196    0.0258    0.1579    0.0269    0.0298    0.1289
    0.0280    0.0415    0.0867    0.0477    0.1369    0.2193    0.1007    0.0023    0.0430    0.0428
    0.1531    0.1300    0.0486    0.2457    0.0509    0.0440    0.1059    0.1232    0.0509    0.0492
    0.0234    0.1717    0.0204    0.0316    0.1053    0.0625    0.1430    0.1909    0.0594    0.0179
    0.0249    0.1534    0.1821    0.0924    0.1683    0.1874    0.0139    0.2105    0.0961    0.2120
    0.1357    0.0153    0.0485    0.0746    0.0755    0.1145    0.0741    0.1547    0.0958    0.1640
    0.0696    0.0648    0.0916    0.1668    0.1213    0.1806    0.1812    0.0317    0.0659    0.1294
    0.1385    0.0653    0.1302    0.0287    0.1323    0.0930    0.1633    0.1840    0.0760    0.0727
    0.1586    0.1212    0.0540    0.1273    0.0755    0.0641    0.0294    0.0368    0.2703    0.0386
    0.1235    0.1058    0.1515    0.0329    0.1144    0.0087    0.0305    0.0391    0.2129    0.1444

ans =

    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000
```

II. 문제 2번 _ iii. 2)코드 작성 과정

우선 $P = \text{rantrans}(n)$ 이라는 함수를 지정해줬습니다. 그리고 (0,1)의 균일 분포 난수를 요소로 하기 때문에, rand함수를 써줬습니다. 1의 조건대로 각 열의 합이 1이어야 하기 때문에 P의 요소를 $\text{sum}(P,1)$ 로 나눠줬습니다. 그리고 transition_matrix라는 스크립트를 작성하여 n을 직접 입력해주고 rantrans함수를 불러와 P를 출력했습니다. 그리고 각 열의 합이 1인지 확인하기 위해 $\text{sum}(P)$ 를 출력하여 1의 조건도 충족 시킴을 확인할 수 있습니다.

II. 문제 2번 _ iv. 3) 결과 스크린 샷

```
>> main
x(1) x(2) x(3) x(4)
-----
1000000 1000000 1000000 1000000
835030 1175151 873190 1116630
804960 1196677 880142 1118221
791231 1204053 881244 1123472
784684 1207415 882922 1124979
781779 1209074 883402 1125746
780434 1209793 883659 1126113
779821 1210129 883778 1126272
779540 1210282 883830 1126347
779412 1210352 883855 1126381
779354 1210384 883866 1126396
779327 1210399 883871 1126404
779314 1210405 883873 1126407
779309 1210408 883875 1126408
779306 1210410 883875 1126409
779305 1210410 883875 1126409
779305 1210411 883875 1126409
779304 1210411 883875 1126409
779304 1210411 883875 1126410
779304 1210411 883875 1126410
steady state: 19years
```

II. 문제 2번 _ v. 3) 코드 작성 과정

4개의 섬 인구를 비교하므로 $n = 4$ 라고 선언해줍니다. 2번에서 만들었던 rantrans함수를 호출하여 P를 선언해줍니다. steady state가 되는 연도를 t라는 변수를 세워 나타내기 위해 선언해줍니다. 기존의 코드에서 for가 아닌 while을 써야 하므로 while에서 x와 y를 비교해주기 위해 $y = \text{Transition}(P, x)$ 라고 선언해줍니다. 그리고 x와 y를 정수형으로 받아 같지 않을 동안 while문을 실행시킵니다. 비교를 위해 $x = y$ 라고 하고, $y = \text{Transition}$ 을 해줍니다. 그리고 y를 출력해주고 t를 1씩 증가시킵니다. 그러다 x와 y가 같아지면 while문을 빠져나와 연도를 출력해줍니다.