In the name of God

استاد : دکتر تیموری

دانشجو : توحید حقیقی سیس

شماره دانشجویی : 830598021

موضوع : تمرین **پنجم**

تمرین اول :

1. Write a program to generate n values from the probability mass function $p_1 = \frac{1}{3}, p_2 = \frac{2}{3}$.

   (a) Let $n = 100$, run the program, and determine the proportion of values that are equal to 1.
   (b) Repeat (a) with $n = 1000$.
   (c) Repeat (a) with $n = 10{,}000$.

در این تمرین به این صورت عمل میکنیم که شروع میکنیم به تولید اعداد تصادفی بین صفر و یک ، اگر از ۰.۳ کمتر بود جز مورد اول و اگر از ۰.۳ بود تا ۰.۳+۰.۷ =۱ نسبت به مورد دوم است .

یک شمارنده برای هر یک قرار میدهیم و تعداد ان ها را میشماریم تا تعداد توزیع آن ها را به دست می آوریم .

در این سوال عدد تصادفی ای تولید می شود و اگر کمتر از ۰.۳ بود مقدار ۱ به ان نسبت و در غیر این صورت مقدار ۲ به ان نسبت داده می شود. در نهایت احتمال ها تولید و چاپ میشوند که نزدیک مقادیر گفته شه هستند که نشان از درستی روش استفاده شده است .

تابع mass_function تابع اصلی من است که از ورودی عدد میگیرد و محاسبات را انجام میدهد .

و خروجب برنامه به ثورت زیر است :

with n = 100 , proportaion of 1: 0.32 proportion of 2: 0.68

with n = 1000 , proportaion of 1: 0.331 proportion of 2: 0.669

with n = 10000 , proportaion of 1: 0.3374 proportion of 2: 0.6626

```python
import math
import numpy
import pandas
import random

#------------ Homework 1 ----------------------
print("--------------------------- homework 1 -------------------------------------")

n_count=int(input(" Get Count Simulation From User : "))

# تابعی که تعداد دو عدد رندوم
# را محاسبه میکند
def mass_function(n):
    #0.3 تعداد کوچک تر از
    count_1=0
    # 1 و 0.3 تعداد بین
    count_2=0
    for i in range(n):
        #تولید عدد تصادفی
        rand_n = random.random()
        if(rand_n<0.3):
            # افزودن تعداد
            count_1=count_1+1
        else:
            count_2=count_2+1

    return count_1,count_2


num_1_100,num_2_100=mass_function(100)
num_1_1000,num_2_1000=mass_function(1000)
num_1_10000,num_2_10000=mass_function(10000)
print("with n = 100 ,proportaion of 1:",num_1_100/100,"proportion of 2:" ,num_2_100/100 )
print("with n = 1000 ,proportaion of 1:",num_1_1000/1000,"proportion of 2:" ,num_2_1000/1000 )
print("with n = 10000 ,proportaion of 1:",num_1_10000/10000,"proportion of 2:" ,num_2_10000/10000 )
```

```
 Get Count Simulation From User : 100
with n = 100 ,proportaion of 1: 0.32 proportion of 2: 0.68
with n = 1000 ,proportaion of 1: 0.331 proportion of 2: 0.669
with n = 10000 ,proportaion of 1: 0.3374 proportion of 2: 0.6626
```

تمرین دوم :

2. Write a computer program that, when given a probability mass function $\{p_j, j = 1, \ldots, n\}$ as an input, gives as an output the value of a random variable having this mass function.

مثل قسمت قبل با این تفاوت که به جای ۲ احتمال تعداد نا مشخص احتمال داریم که جمع انها ۱میشود .

در این سوال ابتدا تعداد و سپس مقادیر احتمال و value مربوطه به ان احتمال دریافت شده سپس مرتب سازی بر اساس مقادیر احتمال ها می شوند حال با جمع مقادیر احتمال ها به صورت تجمعی جدولی بدست می اید که در قسمت بعدی ابتدا در یک حلقه عدد رندومی تولید میکنیم و هر جا عدد رندوم از مقادیر احتمال جمع شده کوچکتر بود ان عدد را به عنوان خروجی بر می گردانیم در واقع عدد رندومی با توزیع دلخواه گفته شده تولید میکنیم که خواسته سوال است .

ابتدا اعداد احتمال را از ورودی از کاربر میگیریم :

```
import math
import numpy as np
import pandas
import random

#-------------  Homework 2  -------------------
print("------------------------- homework 2 ---------------------------------------")

# گرفتن لیست احتمالات از ورودی
def input_dis():
    dis_table = []
    n = int(input("Enter value of n:"))
    for i in range(n):
        x = float(input("probability for spesific value: "))
        dis_table.append(x)
    return dis_table
```

سپس با این اعداد یک ارایه از بازه ها از ۰ تا ۱ تولید میکنیم :

برای این کار ابتدای ارایه را ۰ میگذاریم و انتهای ان را ۱ قرار میدهیم .و هر یک از ورودی ها را با مقدار جمع قبلی ها جمع میکنیم برای مثال اعدا زیر را از ورودی میگیریم :

Input : 0.1 , 0.2 , 0.3 , 0.1 ,0.2,0.1

Array : 0, 0.1,0.3,0.6,0.7,0.9,1

```
#درست کردن لیست محدوده بازه ها که از 0 شروع و با جمع هر عدد تا 1 میرود
def find_sum_array(array):
    list_probability=[]
    list_count=[]
    # اولی رو 0 میگیریم
    sum_probability=0.0
    #به لیست بازه اضافه میکنیم
    list_probability.append(sum_probability)
    # شروع به پیمایش کل اعداد ورودی میکنیم و هر کدام را با جمع مقدار قبلی جمع میکنیم
    for probability in array:
        sum_probability=float(sum_probability + probability)
        list_probability.append(sum_probability)
        list_count.append(0)
    return list_probability,list_count
```

و در مرحله بعد شروع میکنیم به تولید اعداد تصادفی و در هر بازه ای که قرار گرفت آن بازه را +۱ میکنیم .

```
def mass_function(list_array,list_count,n):
    print(n)
    count_list=[]
    for i in range(n):
        rand=float(random.random())
        #5print(rand)
        for j in range(1,len(list_array)):
            #print(list_array[j-1])
            if(rand>float(list_array[j-1])):
                #print(list_array[j])
                if(rand<float(list_array[j])):
                    #print("is ok")
                    #print(list_count)
                    list_count[j-1]=list_count[j-1]+1
    return list_count
```

و طریقه و ترتیب فراخوانی توابع به شکل زیر است :

```python
if __name__ == "__main__":
    input_list=input_dis()
    probability_list,list_count=find_sum_array(input_list)
    print(probability_list)
    print(list_count)
    count_l=mass_function(probability_list,list_count,100)
    print(count_l)
    results=[]
    for k in count_l:
        results.append(k/100)
    print(results)
```

خروجی برنامه به صورت زیر است :

```
------------------------- homework 2 ---------------------------------------
Enter value of n:5
probability for spesific value: 0.1
probability for spesific value: 0.2
probability for spesific value: 0.1
probability for spesific value: 0.3
probability for spesific value: 0.2
[0.0, 0.1, 0.30000000000000004, 0.4, 0.7, 0.8999999999999999]
[0, 0, 0, 0, 0]
100
[11, 20, 7, 29, 23]
[0.11, 0.2, 0.07, 0.29, 0.23]
```

همان طور که از اعداد اخر به دست آمده میبینیم تعداد هر بازه با احتمال آن برابر شبیه سازی شد .

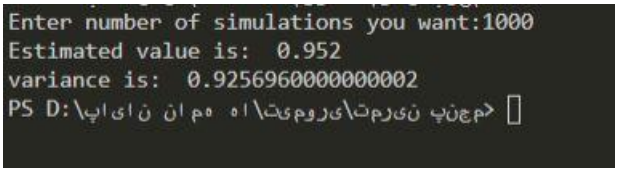
تمرین سوم :

4. A deck of 100 cards—numbered 1, 2, . . . , 100—is shuffled and then turned over one card at a time. Say that a "hit" occurs whenever card i is the ith card to be turned over, $i = 1, \ldots, 100$. Write a simulation program to estimate the expectation and variance of the total number of hits. Run the program. Find the exact answers and compare them with your estimates.

در این سوال به این صورت عمل میکنیم که ابتدا یک ارایه از اعداد ۱ تا ۱۰۰ را تشکیل میدهیم و بعد در هر بار بازی یک بار جاهای ۱ تا ۱۰۰ را به هم میزنیم و شروع میکنیم به شمردن اینکه ایا کدام یک ازخانه ها تغییر جا نداده اند و هنوز در جای خودشان هستند و آن ها را میشمارین این کار را به تعداد زیاد انجام میدهیم و میانگین و واریانس رو پیدا می کنیم .

```
import random
cards = []
sum = 0
ssum = 0

for i in range(100):
    cards.append(i+1)
numofsim = int(input("Enter number of simulations you want:"))
def play():
    hit = 0
    random.shuffle(cards)
    r = random.randrange(0,100)
    # print(len(cards))
    for i in range(len(cards)):
        if cards[i] == i+1:
            hit += 1
    return hit

for i in range(numofsim):
    t = play()
    sum = sum + t
    ssum = ssum + t**2

e = sum/numofsim;
v = ssum/numofsim - e**2;

print("Estimated value is: " ,e)
print("variance is: ",v)
```

خروجی برنامه به صورت زیر خواهد شد :



تمرین چهارم :

7. A pair of fair dice are to be continually rolled until all the possible outcomes 2, 3, . . . , 12 have occurred at least once. Develop a simulation study to estimate the expected number of dice rolls that are needed.

در این سوال این گونه عمل میکنیم که در یک حلقه بینهایت شروع به تولید اعداد تصادفی بین ۱و۶ میکنیم و بعد هر کدام را جمع میکنیم و اگر در ارایه ای که اعداد ۲ تا ۱۲ بود فلگ ان را یک میکنیم .

این حلقه را تا جایی تکرار میکنیم که فلگ همه ۱ شود و تعداد دفعات تکرار را بر میگردانیم .

خروجی به صورت زیر است :

```
76
151
85
89
61
31
113
52
41
41
25
100
34
44
153
59
69
72
83
39
49
78
72
26
50
49
31
64
112
49
82
54
66
51
67.15
```

```python
import random
#------------------------------------ homework 4 ---------------------------------------
print("------------------------- homework 4 ---------------------------------------")

dices = [2,3,4,5,6,7,8,9,10,11,12]
dices_is_exist=[0,0,0,0,0,0,0,0,0,0,0]

def check_array():
    for dices in dices_is_exist:
        if(dices==0):
            return True
    return False

def find_count_dices():
    counter=0
    while(1==1):
        counter=counter+1
        if(check_array()):
            rand_1=random.randint(1,6)
            rand_2=random.randint(1,6)
            final_rand=rand_1 + rand_2
            for i in range(len(dices)):
                if(final_rand==dices[i]):
                    dices_is_exist[i]=1
        else:
            break
    return counter

count_of_simulation=100
sum_of_simulation=0
for i in range(count_of_simulation):
    sum_of_simulation=sum_of_simulation+int(find_count_dices())

print(sum_of_simulation/count_of_simulation)
```