Lab Number : 03
Due Date : Sept 23, 2020
Student Details :

- Name : AB Satyaprakash
- Roll Number : 180123062
- Department : Mathematics and Computing

Question 1 :

Observations :

a) For this question we first had to generate the value of q-array. Since the discrete value are uniform, this means each has a probability (call P(i)), **P(i)=1/5000** (as there are a total of **5000** discrete values)

b) Q(i) is generated by a simple recursion, **Q(0)=0**, and **Q(i)=Q(i-1)+P(i)**

c) This done, we have made 2 cases for 10 and 100 values of U.
The U ~ U [0,1] is generated via linear congruence generator for 100 values with the following seed :
x0=23, **m**=4096, **a**=17, **b**=1
{ X(i+1)=(X(i)A + B) mod M and U(i+1)=X(i+1)/M } -> Linear Congruence Generator

d) In each case for U, we verified for all values of U if there is a k, such that Q(k-1)<U<=Q(k). If this holds then we returned C(k) as the corresponding random number.

e) In 2 cases we found 10 and 100 random numbers that are listed below.

Output :

```
This is for the case of 10 values of U

[59, 959, 6273, 6627, 2647, 4953, 4175, 935, 5859, 9573]


This is for the case of 100 values of U

[59, 959, 6273, 6627, 2647, 4953, 4175, 935, 5859, 9573, 2699, 5867, 9697, 4817, 1851,
1423, 4159, 685, 1625, 7605, 9247, 7179, 2025, 4411, 4961, 4299, 3051, 1843, 1299,
2043, 4701, 9899, 8261, 413, 6977, 8581, 5849, 9407, 9877, 7887, 4063, 9025, 3403,
7819, 2899, 9269, 7553, 8377, 2363, 139, 2329, 9557, 2451, 1633, 7729, 1365, 3163,
3753, 3755, 3795, 4501, 6497, 405, 6853, 6465, 9865, 7681, 535, 9053, 3859, 5581, 4841,
2265, 8479, 4107, 9773, 6103, 3723, 3257, 5329, 565, 9591, 3031, 1511, 5653, 6085,
3433, 8317, 1367, 3205, 4457, 5749, 7705, 949, 6107, 3805, 4667, 9319, 8383, 2487]
```

Question 2 :

Observations :

a) $f(x) = 20x(1-x)^3$ and the known density function is g, which is $\mathcal{U}[0,1]$.
We need to find the smallest constant c, here which satisfies the inequality, $f(x) \le cg(x)$, since g is the uniform function, we have to find c, such that, $c \ge f(x)/g(x) \Rightarrow c \ge f(x)$. Now this means we need the maximum value of f(x) at any given x.

Differentiating wrt to x, we have $f'(x) = 20(1-x)^3 + 60x(1-x)^2$. Equating RHS to 0, we obtain x=¼. Substituting this in f(x), we have the maximum value of f(x)=2.109375.

Thus the smallest constant c which satisfies the inequality is 2.109375.

b) Using the c we found above, we can generate random values via the acceptance rejection method. I did this for 100,1000,10000 and 100000 random values and in each case, I found the U, X and computed f(X). Then if the value was >= U, we accepted the value or rejected it.
The count for the number of iterations was also stored. The whole process can be explained in the small code segment below.

To check if the values converge, we used **plt.hist()** to generate the corresponding frequencies and then used the following regression formula to calculate the error between f(x) and the frequency values.

$$error = (f(x) - y)^2/num$$

where num represents the total number of values of X *that we generated.*
As we can see from the output section, the error is quite insignificantly small, thus the values converged.

```
def get_iter_forx(num,c): # we are taking num random values at a time
    itr_for_x=[]
    for i in range(0,num):
        count=1
        while(1):
            u=random.uniform(0,1) # represents U ~ U[0,1]
            x=random.uniform(0,1) #represents the X generated from g, which is
U[0,1]
            fx=20*x*(pow(1-x,3)) # f(x) is calculated by 20*x*(1-x)^3
            if(u<=(fx/(c))):
                break
            else:
                count=count+1
        itr_for_x.append(count)
```

c) The average of all the counts of the number of iterations was computed and in each case we found the average to be **very close to the value of c** that we have taken. The same can be seen from the output section.
d) We took the following 2 values of c, (that is c+1 and c+2):
    - c=3.109375
    - c=4.109375

We observe that the **values of average also converge to c in each case**. However one interesting observation that can be gotten is that the **count of the number of iterations increases with increase in c**. In other words, the higher value of c gives more rejections than acceptance. The explanation is simple, since we have to make $f(x)/cg(x) >= U$, with increasing c, we will have more cases of $f(x)/cg(x) < U$. As per the Law of Large Numbers, on a large sample we have that the average of the number of iterations is very close to the expected value of the number of iterations.

Output :

```
The value of c used is 2.109375
The mean value of count of number of iterations for 100 random values is 2.26
The error value is 0.2666017068167283
The mean value of count of number of iterations for 1000 random values is 2.129
The error value is 0.03446949829597629
The mean value of count of number of iterations for 10000 random values is
2.1128
The error value is 0.02082403559607948
The mean value of count of number of iterations for 100000 random values is
2.11024
The error value is 0.022116133923678612


The value of c used is 3.109375
The mean value of count of number of iterations for 100 random values is 3.22
The error value is 0.24457700921956974
The mean value of count of number of iterations for 1000 random values is 2.975
The error value is 0.05467297024385619
The mean value of count of number of iterations for 10000 random values is
3.1351
The error value is 0.025940497208931202
The mean value of count of number of iterations for 100000 random values is
3.10761
The error value is 0.02238008590538678


The value of c used is 4.109375
The mean value of count of number of iterations for 100 random values is 3.62
The error value is 0.43689159142276673
The mean value of count of number of iterations for 1000 random values is 3.889
The error value is 0.04654642981316923
The mean value of count of number of iterations for 10000 random values is 4.147
The error value is 0.023824061738145635
The mean value of count of number of iterations for 100000 random values is
4.1041
The error value is 0.021872030708267202
```

Question 3 :

Observations :

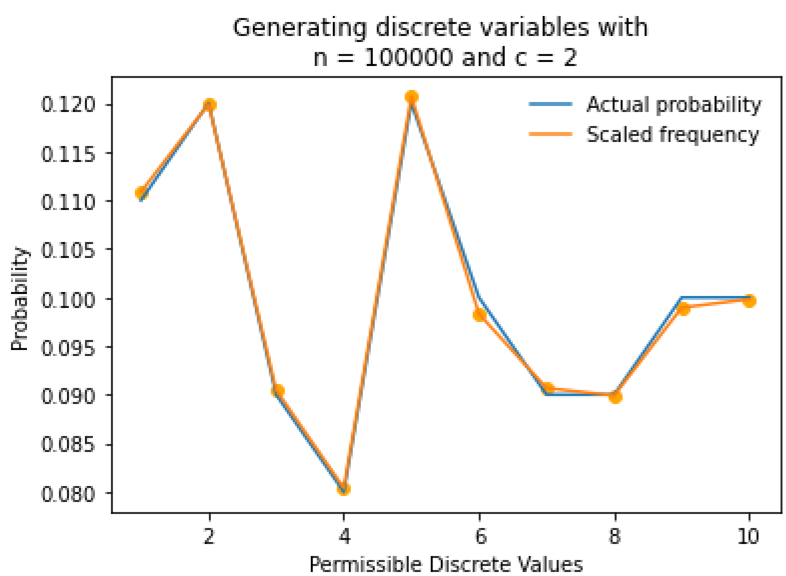
Since the base distribution is a discrete distribution on the set A={1,2,3,..,10}, we have the PMF=0.1, or g(x)=0.1 for $\forall\ x \,\varepsilon\, A$. Also since f(x) takes a maximum value of 0.12, the minimum value for c to take will be 1.2. So, I have taken the values of c to be 2 and 3 in my case.

I generated a sequence of variables X with 1,00,000 elements for both the values of c and plotted the values against their scaled frequency of occurrence.

As we can observe the acceptance rejection method gives almost perfect results for the large number of observations generated. In this case too, the mean of the iterations required matches the value of c used.

Output :

```
For the values of n=100000 and c=2
We have the mean=2.0056
```




```
For the values of n=100000 and c=3
We have the mean=2.98619
```