ASSIGNMENT 7

MA226 : Monte Carlo Simulation
Name: Nikhil Agarwal
Roll No : 11012323
IIT Guwahati

---

1 Problem 1

1.1 Statement

1. Generate 50 random numbers from geometric distribution of the form $f(x, p) = pq^{i-1}$ where $i = 1, 2, ...0 < p < 1..$ Draw the probability mass function .

1.2 Solution

1.2.1 Graph

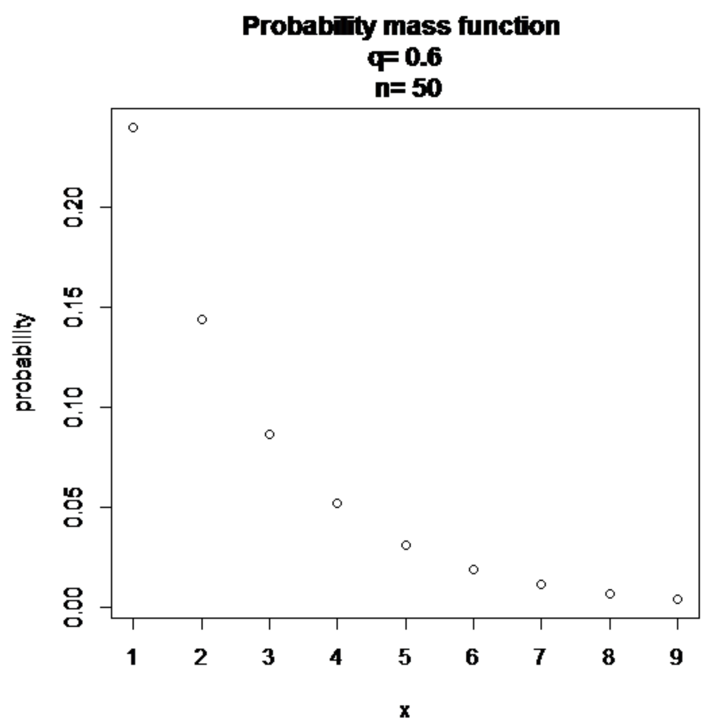

Figure 1: Probability mass function for p=0.4

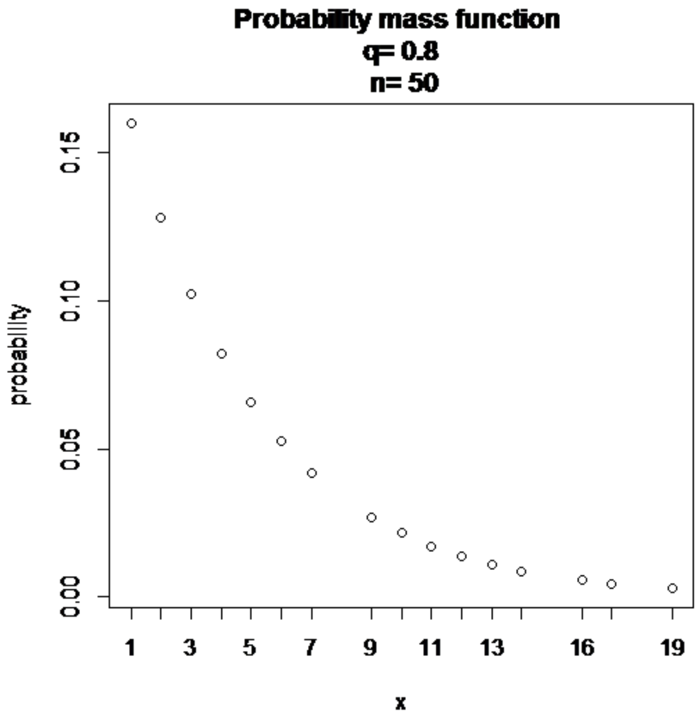

Figure 2: Probability mass function for p=0.2

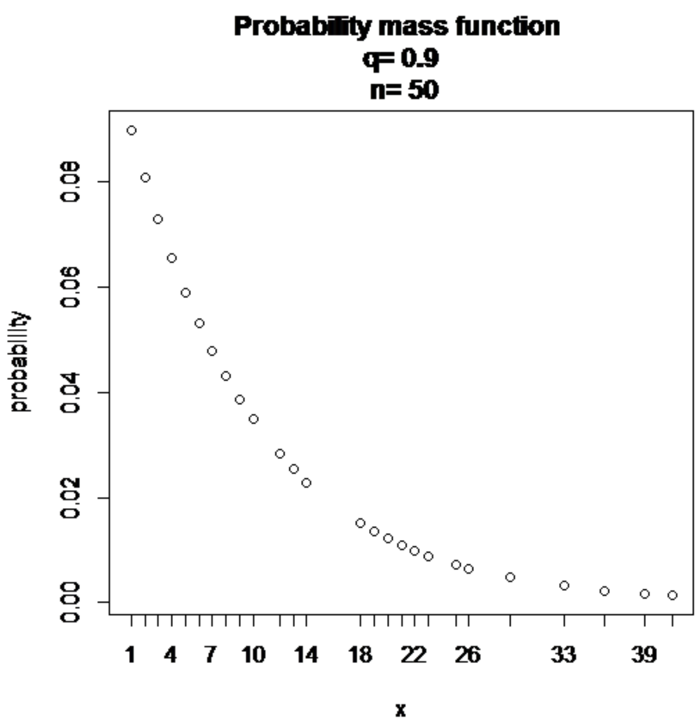

Figure 3: Probability mass function for p=0.1

1.2.2 R Code

```
u<-vector()
x<-vector()
v<-vector()
geometric<-function(m)
{
    q=0.6
    set.seed(6857)
    u<-runif(m)
    x<-floor(log(u)/log(q))+1
    print(x)
    v<-(1-q)*(q^x)
    plot(x,v,"p",xlab="x",ylab="probability",xaxt="n",main=paste
    ("Probability mass function","\n","q=",paste(q),"\n","n=",paste(m)))
    axis(side = 1, at = x)
    #par(new=TRUE, col="yellow")
    print(q)
}
geometric(50)
x11()
geometric(20000)
```

Listing 1: R Code which generates random numbers from Geometric distribution

1.2.3 Results

As the value of p decreases we get more and more distinct values of x. As, p decreases , q increases and pq^{i-1} gives larger value for small values of i because $q < 1$ hence we get larger probabilities for smaller values of i .

2 Problem2

2.1 Statement

2. Generate 50 random numbers from poisson distribution with mean 2. Draw the probaability mass function and the cummulative distribution function.

2.2 Solution

2.2.1 Graph

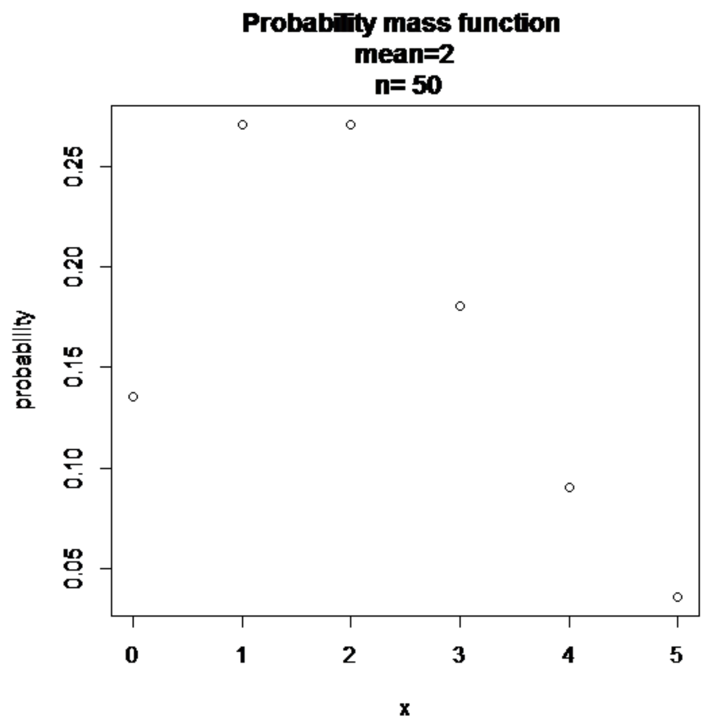

Figure 4: Probability Mass function for n=50

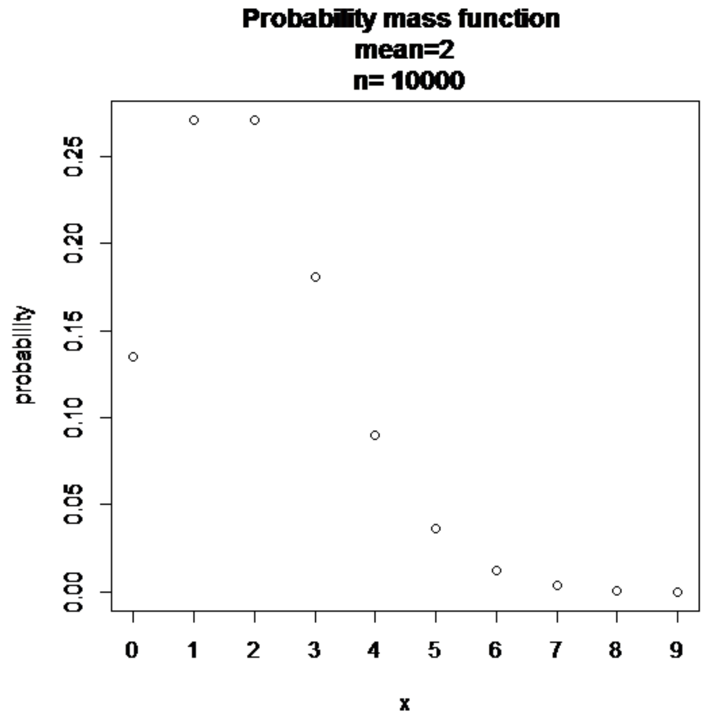

Figure 5: Probability Mass function for n=10000

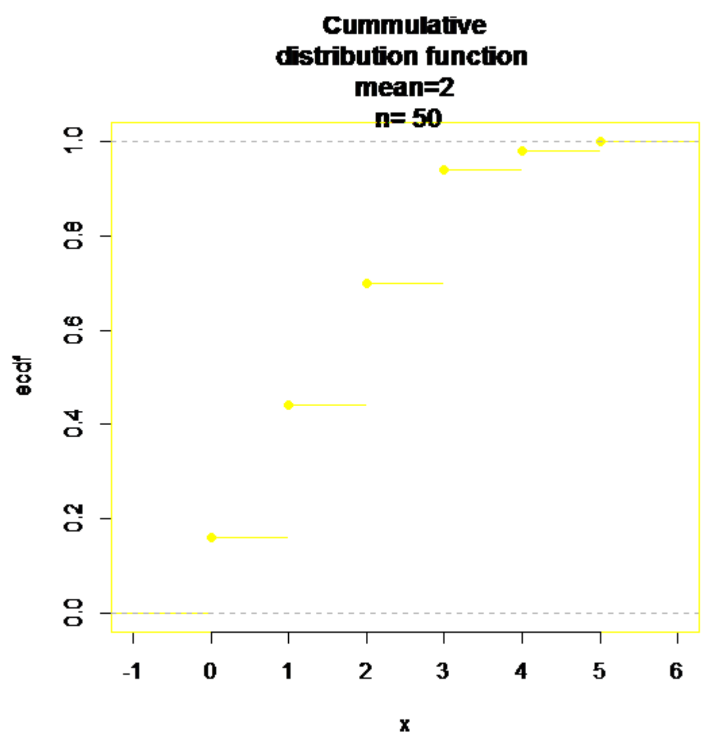

Figure 6: Cummulative distribution Function for n=50

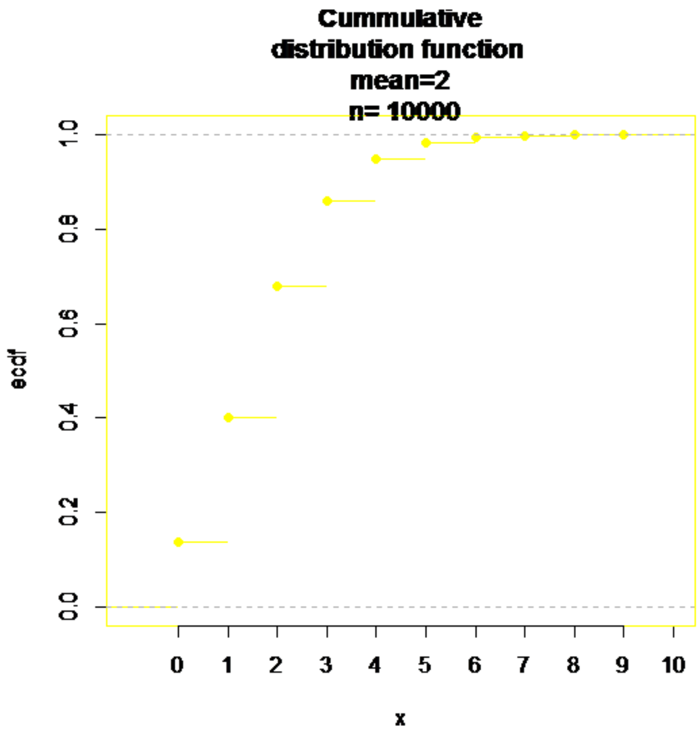

Figure 7: Cummulative distribution Function for n=10000

2.2.2 R Code

```
x<-vector()
v<-vector()
u<-vector
poisson<-function(n)
{
    set.seed(343219)
    u=runif(n)
    for(j in 1:n)
    {
        i<-0
        p=exp(-2)
        f=p
        m=0
        while(m!=1)
        {
            if(u[j]<f)
            {
                x[j]=i
                m=1
            }
            else
            {
                p=2*p/(i+1)
                f=f+p
                i=i+1
            }
        }
    }
    print(x)
    print(var(x))
    v<-exp(-2)*(2^x)/factorial(x)
    print(v)
    plot(x,v,"p",xlab="x",ylab="probability",xaxt="n",main=paste
    ("Probability mass function","\n","mean=2","\n","n=",paste(n)))
    axis(side = 1, at = x)
    x11()
    par(col="yellow")
    plot(ecdf(x),xlab="x",ylab="ecdf",main=paste("Cummulative
    distribution function","\n","mean=2","\n","n=",paste(n)))
    axis(side = 1, at = x)

}
poisson(50)
#x11()
#poisson(10000)
```

Listing 2: R Code which generates random numbers from Poisson Distribution

2.2.3 Result

The variance of the above distribution came out to be 1.98224 which is quite nearer to 2. Hence generated random numbers are correct and they follow poisson distribution .

3 Problem3

3.1 Statement

3. Draw the Histogram based 50 generated random numbers from the mixture of two Weibull distributions

$$f(x,\beta_1,\theta_1,\beta_2,\theta_3) = pf_1(x,\beta_1,\theta_1) + (1-p)f_2(x,\beta_2,\theta_2)$$

where $f_1(.)$ and $f_2(.)$ are two weibull distributions of the form : $f(x,\beta,\theta) = \beta\theta^{\beta}x^{\beta-1}e^{-\theta x^{\beta}}$ where $\beta_1 = 2, \theta_1 = 1, \beta_2 = 1.5, \theta_2 = 1, p = 0.4$

3.2 Solution

3.2.1 Procedure

- The first weibull distribution function is $1 - e^{-x^2}$ and the second one came out to be $1 - e^{-x^{3/2}}$
- Taking these distributions equal to U(0,1) we get two distributions to be $F_1 = \sqrt{-log(1-u)}$ and $F_2 = (-log(1-u))^{2/3}$
- If p<0.4 we accept F_1 otherwise F_2 and continuing this for 50 numbers , we get a mixed distribution.

3.2.2 Graph

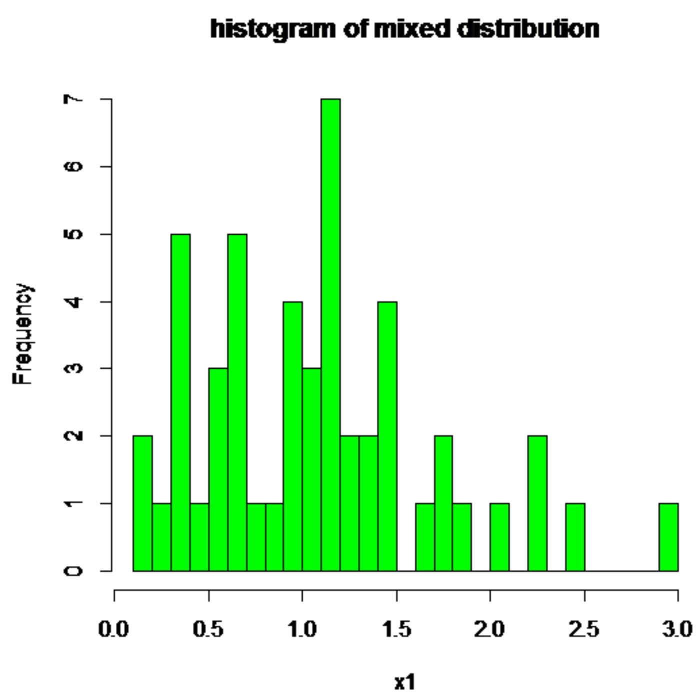

Figure 8: Probability Mass function for n=50

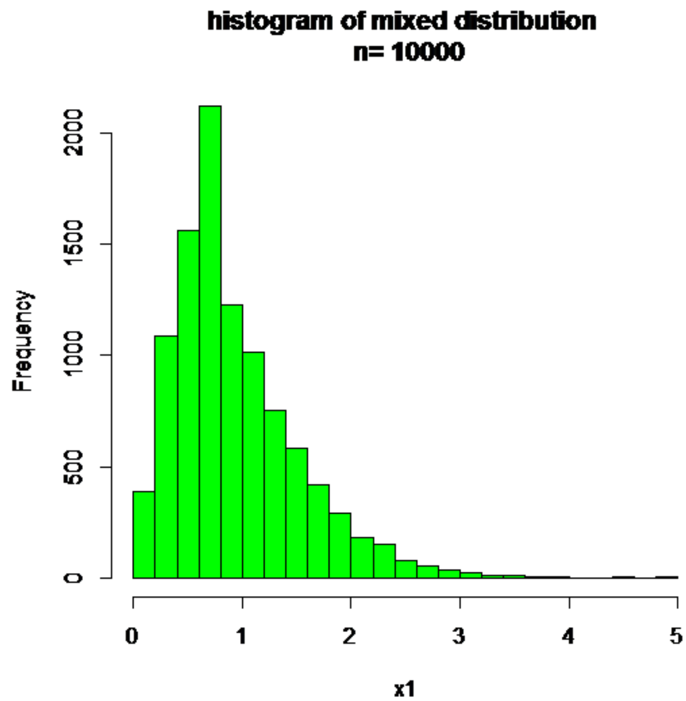

Figure 9: Probability Mass function for n=10000

3.2.3 R Code

```
u<-vector()
x1<-vector()
x2<-vector()
mixed<-function(n)
{
    set.seed(2011)
    u=runif(n)
    j<-1
    k<-1
    mean<-0
    for(i in 1:n)
    {
        if(u[i]<0.4)
        {
            x1[j]=sqrt((-1)*log(1-u[i]))
            mean<-mean+x1[j]
            j<-j+1
        }
        else
        {
            x2[k]=((-1)*log(1-u[i]))^(2/3)
            mean<-mean+x2[k]
            k<-k+1
        }
    }
    for(i in j:n)
    {
        x1[i]=x2[i+1-j]
```

```
    }
    mean<-mean/n
    print(x1)
    print(mean)
    print(var(x1))
    hist(x1,breaks=20,col="green",border="black",
    main=paste("histogram of mixed distribution","\n","n=",paste(n)))
}
mixed(50)
```

Listing 3: R Code which generates random numbers from Mixed Distribution

3.2.4 Result

The mean and variance came out to be 0.892 and 0.267 respectively.