Name: Sourav Bikash

Roll No.:11012338

Submission Date: 08/03/2013 Time:23:59 hrs.

Aim of the Problem:

The problem involves the generation of discreet distributions namely geometric, Poisson and Weibull distributions.

Mathematical Analysis/Theory:

Generating from geometric distribution:

It can be shown that with a random number U, then

$$X=\text{Int}(\log(U)/\log(q)) + 1.$$

is indeed geometric with parameter p.

Thus using U we can generate X following geometric distribution.

Generating from Poisson distribution:

For the case of the Poisson, we exploit the recursion property
$p_{i+1} = (\lambda/(i+1))p_i$ for $i \geq 0$.

The following steps can then be followed to generate from a Poisson with parameter λ:

step 1: generate a random number U.

step 2: set $i = 0$, $p = e^{-\lambda}$ and $F = p$.

step 3: if $U < F$, set $X = i$ and STOP.

step 4: set $p = \lambda p/(i+1)$, $F = F + p$, and $i = i + 1$.

step 5: return to step 3.

The composition of two distributions:

Consider now simulating from a distribution with mass function

$$P(X = j) = \alpha p_j^{(1)} + (1-\alpha)p_j^{(2)} \ ; \ j \geq 0; \ 0 < \alpha < 1$$

If X1 and X2 are the random variables with respective mass functions , then $p_j^{(2)}$ and $p_j^{(1)}$

$X = X1$ with probability α
$X2$ with probability $1 - \alpha$

One approach then to generate from this mixture distribution is :

step 1: generate a random number U1

step 2: generate from X1 and X2 distributions.

step 3: if $U < \alpha$; set X = X1.

step 4: else if $U > \alpha$, set X = X2;

Part I:

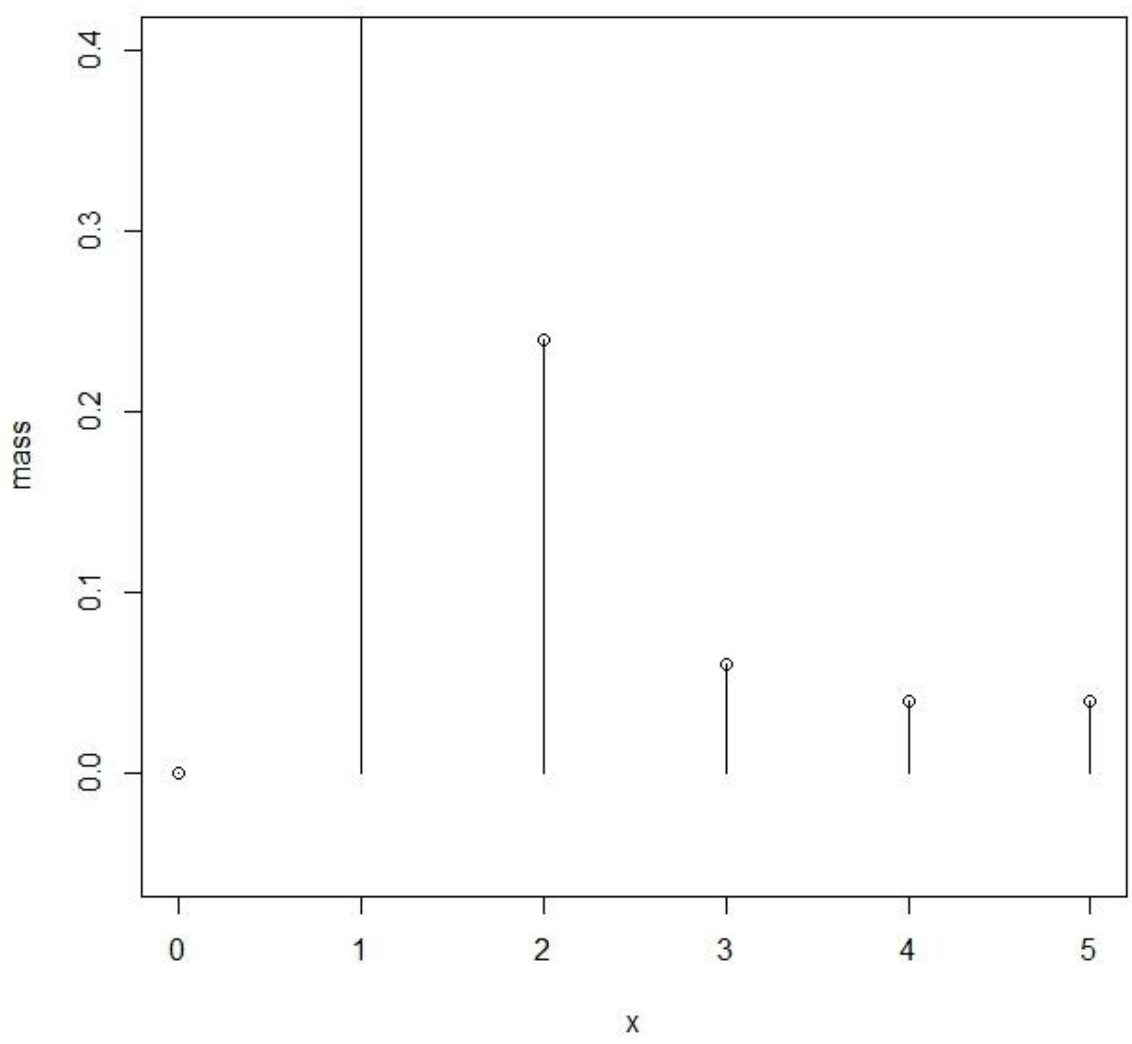
This question wants us to generate 50 random numbers following geometric distribution. For this we take p=0.4.

Implementation using R:

```
geomdist<-function(n)

{

	RN<-NULL;

	for(i in 1:n)

	{

		u1<-runif(1,min=0,max=1);

		p<-0.4;#taking p=0.4

		q<-1-p;

		x<-floor(log(u1)/log(0.4))+1;

		RN<-c(RN,x);

	}

	return(RN);

}

s<-geomdist(50);


mass <- NULL;

for(i in 0:max(s))
```

```
{
        m <- sum(as.integer(i == s));
        mass <- c(mass,m);
}
mass <- mass/50;
x <- seq(0,max(s));
```

Using the output the following mass function generated:



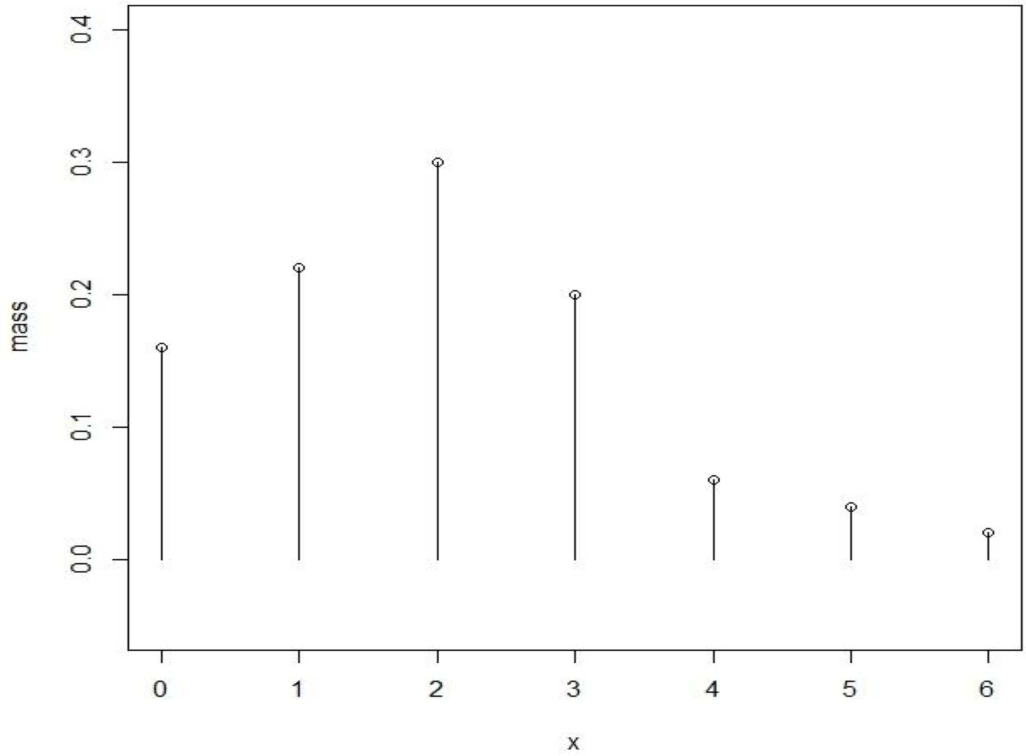
Part II:

Here we need to generate 50 random numbers following poisson distribution.

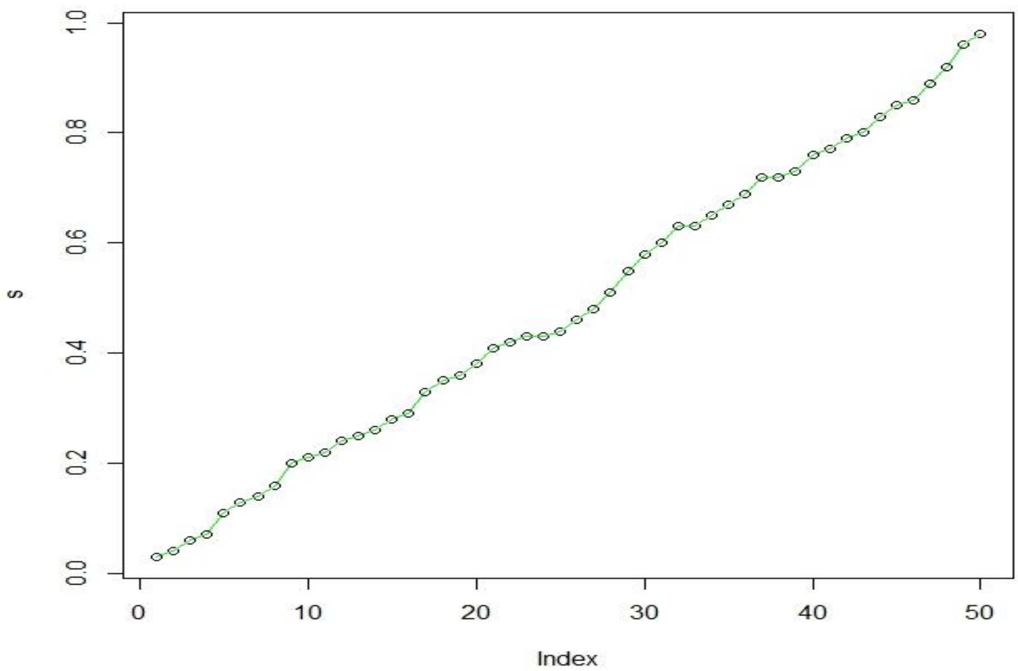
Implementation using R:

```
poisson<-function(n)
{
	RN<-NULL;
	for(i in 1:n)
	{
		u1<-runif(1,min=0,max=1);
		i<-0;
		p<-exp(-2);#mean=2
		f<-p;
		#x<-i;
		while(u1>f)
		{
			p<-p*2/(1+i);
			f<-f+p;
			i<-i+1;
		}
		x<-i;
		RN<-c(RN,x);
	}
	return (RN);
}
s<-poisson(50);
mass <- NULL;
for(i in 0:max(s))
{
	m <- sum(as.integer(i == s));
	mass <- c(mass,m);
}
mass <- mass/50;
x <- seq(0,max(s));
```

Using the output the following mass function was generated:



The cumulative distribution of the numbers was like this:



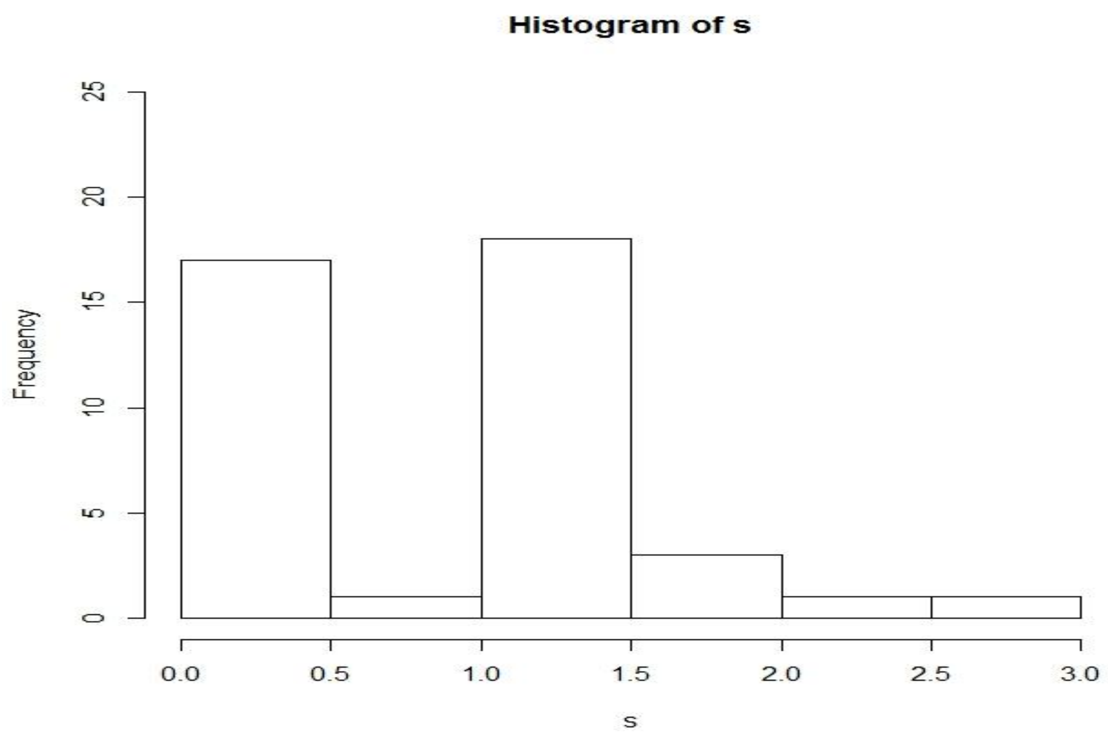
Part III:

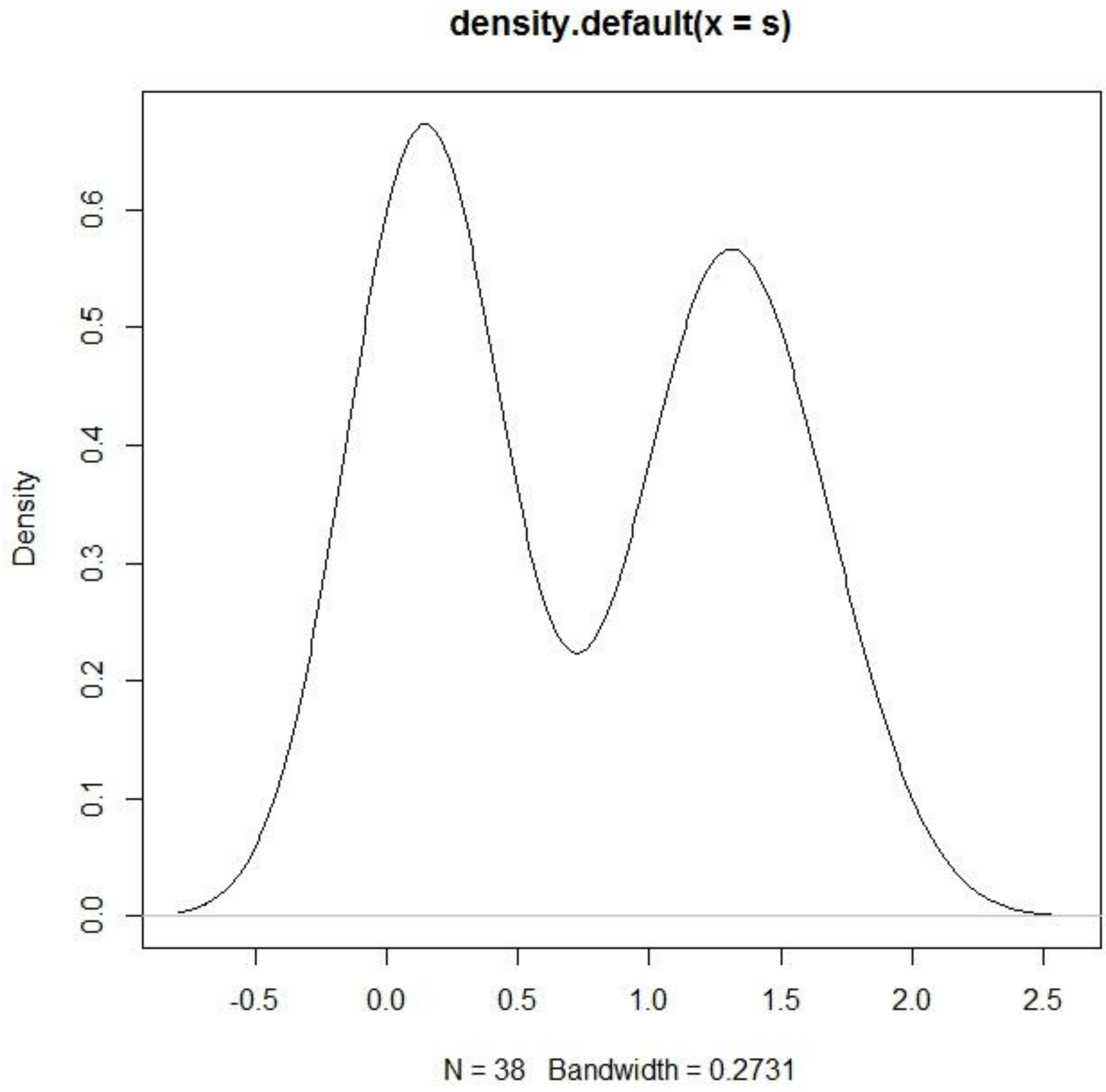
This question asks us to generate joint-Weibull distribution.

The R code:

```
weibull<-function(n){
	RN<-NULL;
	for(i in 1:n)
	{
		u<-runif(1,min=0,max=1);
		x1<-sqrt(log(1/u));
		x2<-(sqrt(log(1/u)))^3;
		if(u<=0.4)
		{
			x<-x1;
			RN<-c(RN,x);
		}
		if(u>0.6)
		{
			x<-x2;
			RN<-c(RN,x);
		}
	}
	return(RN);
}
s<-weibull(50);
```

The output was used to generate a histogram:




density.default(x = s)
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
-0.5
0.0
0.5
1.0
1.5
2.0
2.5
N = 38 Bandwidth = 0.2731