Lab 3

By : Eswar Naveen

Problem 1

Use the Acceptance–Rejection method to generate samples from Binomial(n, p) setting Geometric as the proposal distribution.

(1) For (n, p) = (10, 0.3) use Geometric(p) as the proposal.

(2) Estimate the value of c.

(3) For (n, p) = (100, 0.3) use Geometric($p*$) as the proposal with appropriate choice of $p*$.

(4) Estimate the value of c and compare it with the value of c when the proposal is set as Geometric(p).

Defining required functions

```
#Returns pmf of binomial distribution
binom<- function(n,p,x){
  return (choose(n,x)*(p^x)*(1-p)^(n-x))
}
```

```
#Returns pmf of geometric distribution
geometric<-function(x,p){
  return (p*((1-p)^(x)))
}
```

```
#Returns theoretical value of c
theoretic_c <- function(n,p){
  x_range <- 0:n
  return(max(binom(n,p,x_range)/geometric(x_range,p)))
}
```

```
#Returns samples of binomial(target distribution) using geometric(proposal).
```

```
sample_generator<- function(n,p,c,n_samples,p_geo){
  samples <- numeric()
  while(n_samples){
    U <- runif(1,min=0.0,max=1.0)
    Y <- rgeom(1,p_geo)
    if(U<=binom(n,p,Y)/(c*geometric(Y,p))){
      samples <- c(samples,Y)
      n_samples <- n_samples - 1
    }
  }
  return (samples)
}



#calculates the average amount of guesses required to get a sample.
#which is indeed the value of c
c_calculator<- function(n,p,c1,x){
  count <- 0
  count_it <- numeric(x)
  for(i in 1:x){

    while(TRUE){
      count <- count +1
      U <- runif(1,min=0.0,max=1.0)
      Y <- rgeom(1,p)
      if(U<=binom(n,p,Y)/(c*geometric(Y,p))){
        count_it[i] <- count
        count <- 0
        break
      }
    }
  }
  return (mean(count_it))
}
```

(1) & (2)

Theoretical and estimated values of c.

```
num <- 10
prob <- 0.3

c <- theoretic_c(num,prob)
c
```

```
[1] 2.7783
```

```
#Now lets try to estimate the c value practically
num_samples <- 1e5
c_calculator(num,prob,c,num_samples)
```

```
[1] 2.77153
```

(3)&(4)

Value of c(using geometric(p*))

Here p* = 1/(1+np)

```
n1 <- 100
p1 <- 0.3
p_star = 1/(1+n1*p1)

c_star <- theoretic_c(n1,p_star)
c_star
```

```
[1] 7.715878
```

Value of c(using geometric(p))

```
theoretic_c(n1,p1)
```

```
[1] 48913.21
```