Discrete Probability Distributions

Anirban Mondal

01/13/2021

Binomial Distribution

Find probabilities related to this distribution Example: Let X be binomial random variable with n=10 and p=0.5.

```
?dbinom ## pmf for binomial distribution
n=10
p=0.5
dbinom(2,n,p) ## Finds P(X=2)
```

```
## [1] 0.04394531
```

```
pbinom(2,n,p) ## Finds P(X <= 2) = P(X=0)+P(X=1)+P(X=2)
```

```
## [1] 0.0546875
```

```
mu = n*p ## mean
s2 = n*p*(1-p) ## variance
mu
```

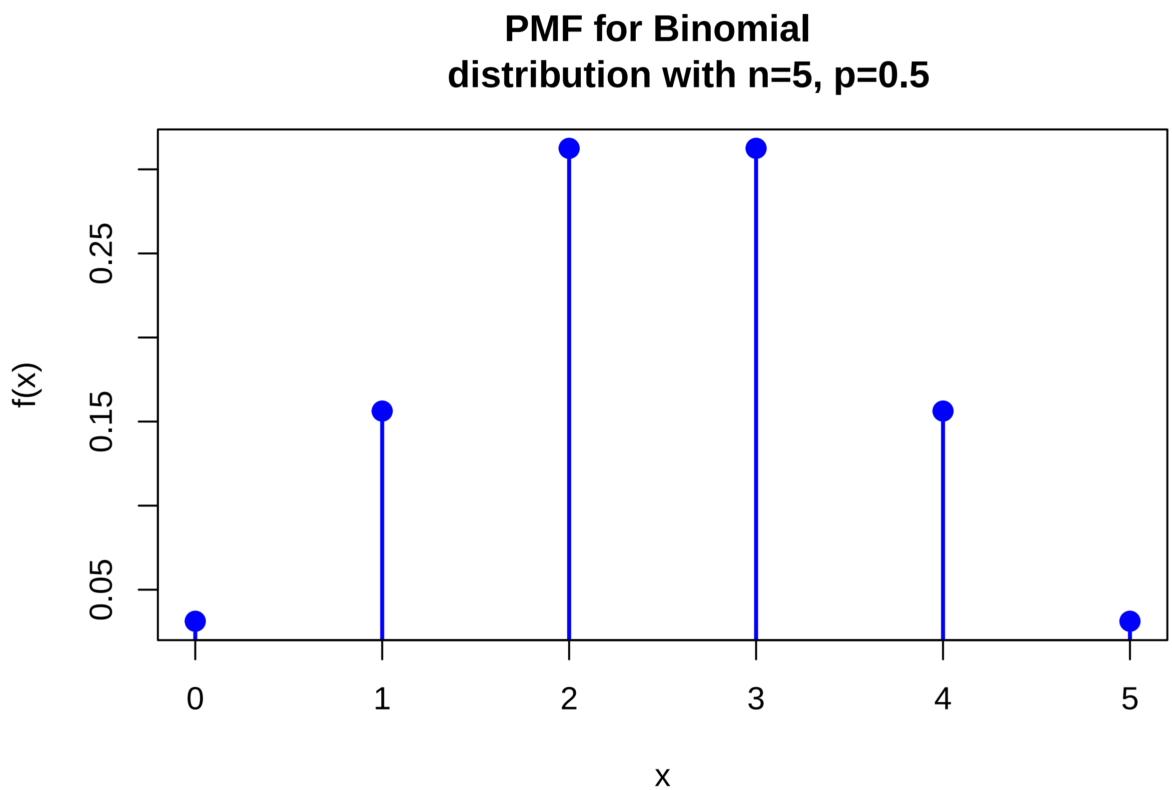
```
## [1] 5
```

```
s2
```

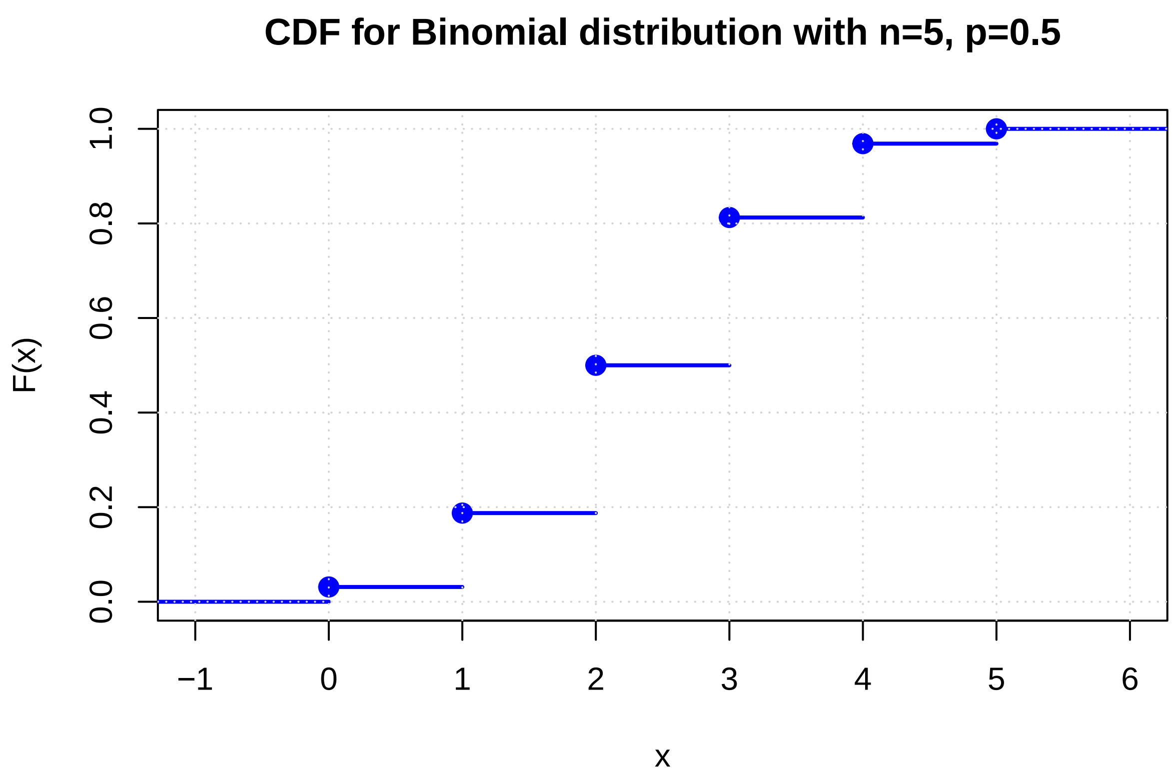
```
## [1] 2.5
```

Plotting the pmf and cdf of Binomial distribution

```
n=5
p=0.5
xx=0:n
## Plotting pmf
plot(xx,dbinom(xx,n,p),type="h",col="blue",lwd=2,main="PMF for Binomial
     distribution with n=5, p=0.5",xlab="x",ylab="f(x)")
points(xx,dbinom(xx,n,p), pch=20, cex=2, col="blue")
```




```
## Plotling CDF
yy=c(0,pbinom(xx,n,p))
plot(stepfun(xx,yy),vertical=F, pch=20, cex=2, lwd=2, col="blue", ylab="F(x)",
     main ="CDF for Binomial distribution with n=5, p=0.5")
grid()
```

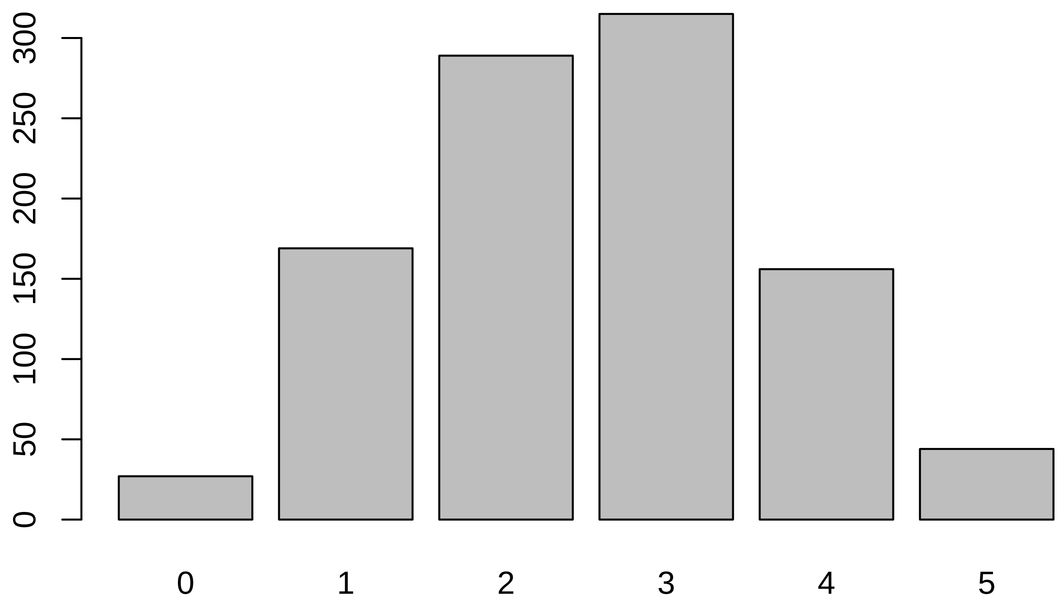


Simulation from binomial distribution

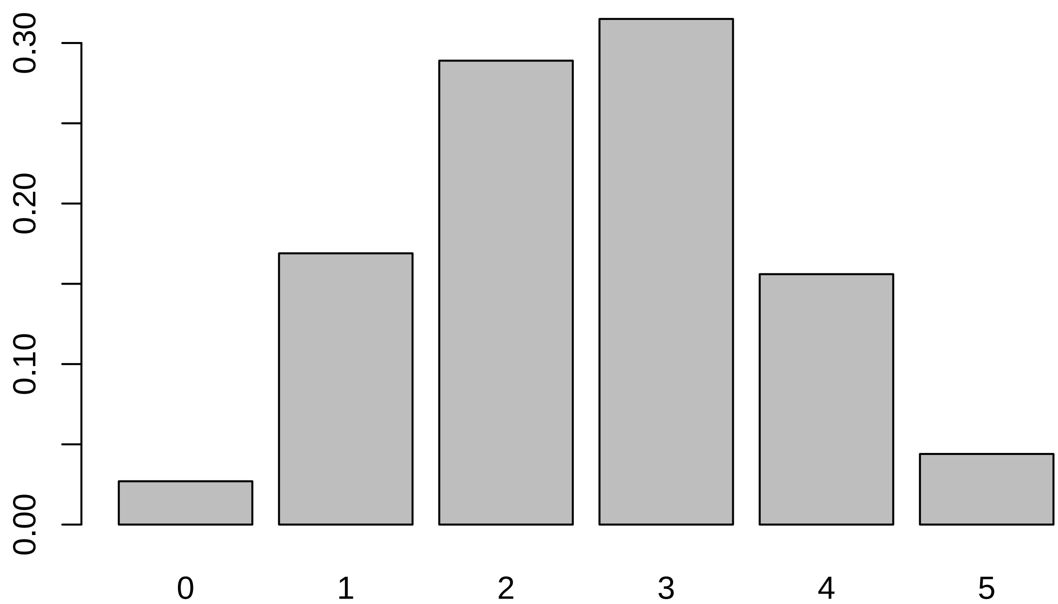
```
rbinom(10,n,p) # Geneartes 10 random samples from a binomial distribution
```

```
##  [1] 1 4 1 2 4 4 3 5 2 4
```

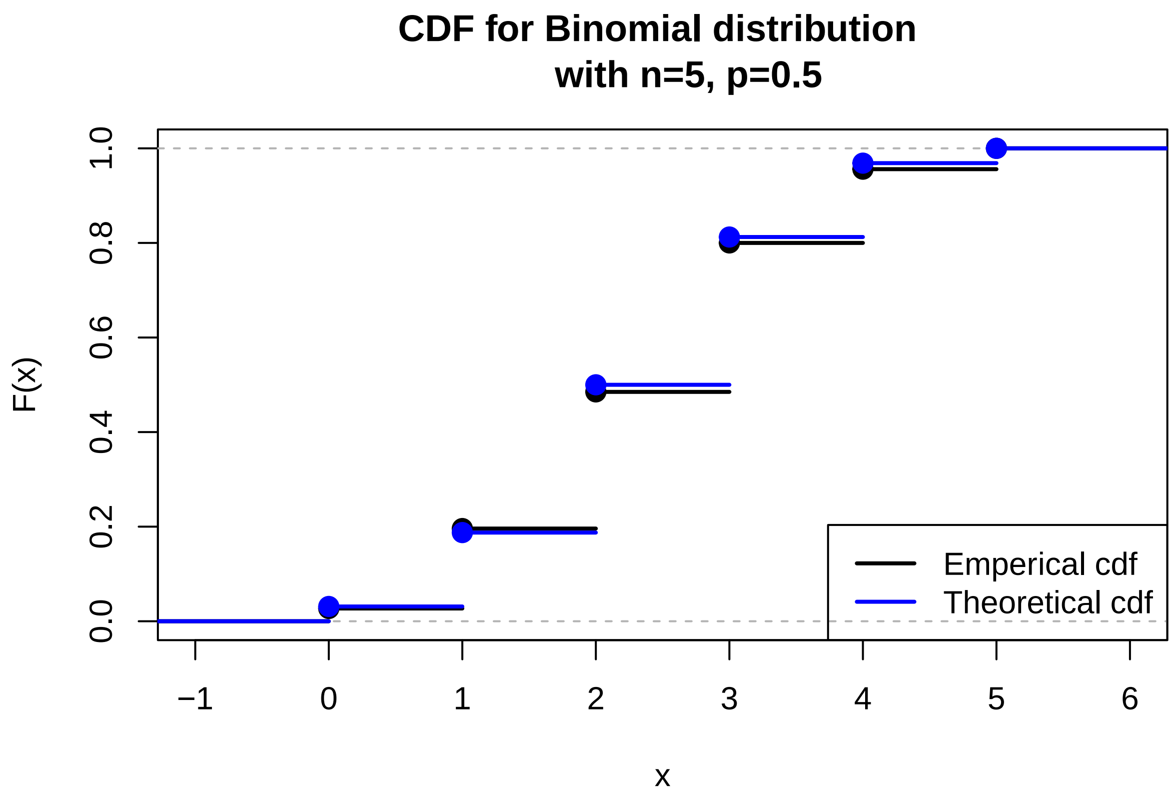
```
ns=1000 ## sample size
set.seed(20) ## setting seed fixes the random number generator
x= rbinom(ns,n,p)
tx=table(x) ## Frequency table for the sample
barplot(table(x)) ## barplot for the frequency
```




```
barplot(prop.table(table(x))) ## barplot using probabilities
```




```
### Plotting emperical cdf of the simulated data and the theoretical cdf overlay
## ecdf (F_n(x)): (Number of x_i <=x)/ns
plot(ecdf(x), lwd=2, pch=20, cex=2, main ="CDF for Binomial distribution
     with n=5, p=0.5", ylab="F(x)")
yy=c(0,pbinom(0:n,n,p))
lines(stepfun(0:n,yy),vertical=F, pch=20, cex=2, lwd=2, col="blue")
legend("bottomright", c("Emperical cdf", "Theoretical cdf"), lty=c(1,1),
       lwd=c(2,2), col=c("black", "blue"))
```



Quantile function for binomial distribution Quantile Definition: $Q(r) = inf\{x \in R : F(x) \geq r\}$

```
qbinom(0.1, n,p) ## Finds the value smallest x such that P(X<=x)>=0.1 (quantile)
```

```
## [1] 1
```

```
pbinom(2,n,p)
```

```
## [1] 0.5
```

```
pbinom(3,n,p)
```

```
## [1] 0.8125
```

Geometric distribution

Note that the geometric distribution in R is defined as the number of failures, not the number of trails. X: no of failures before getting the 1st success. X=Y-1, where Y is the number of trails to get the first success. E(X)=E(Y)-1=1-1/p, V(Y)=V(X) Example: X follows a gemetric distribution of success probablity p=0.2

```
?dgeom
p=0.2
dgeom(3,p) ## P(X=3)
```

```
## [1] 0.1024
```

```
pgeom(2,p) ## P(X<=2)
```

```
## [1] 0.488
```

```
rgeom(10,p) ## Generate 10 random samples from geometric distribution
```

```
##  [1]  2  1 15  1  0  1  1 13  1  0
```

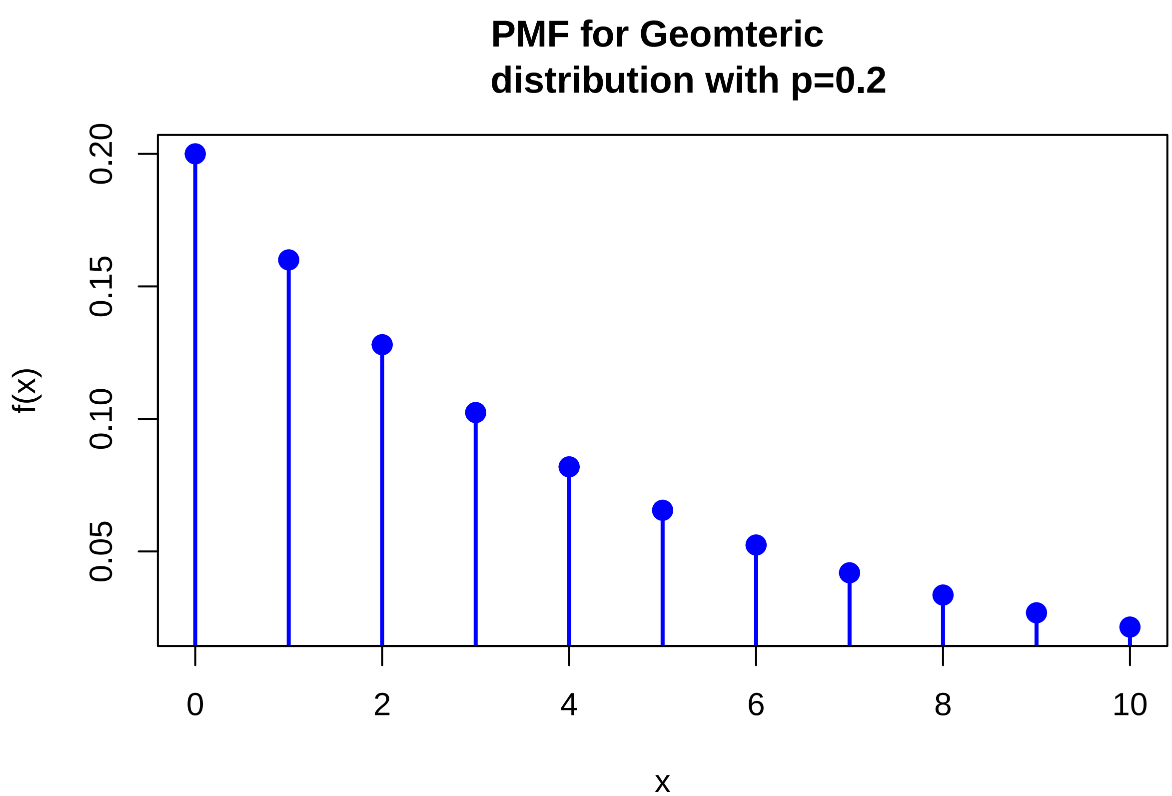
```
qgeom(0.25, p) ## 25th quantile or 1st quartile of the geometric distribution
```

```
## [1] 1
```

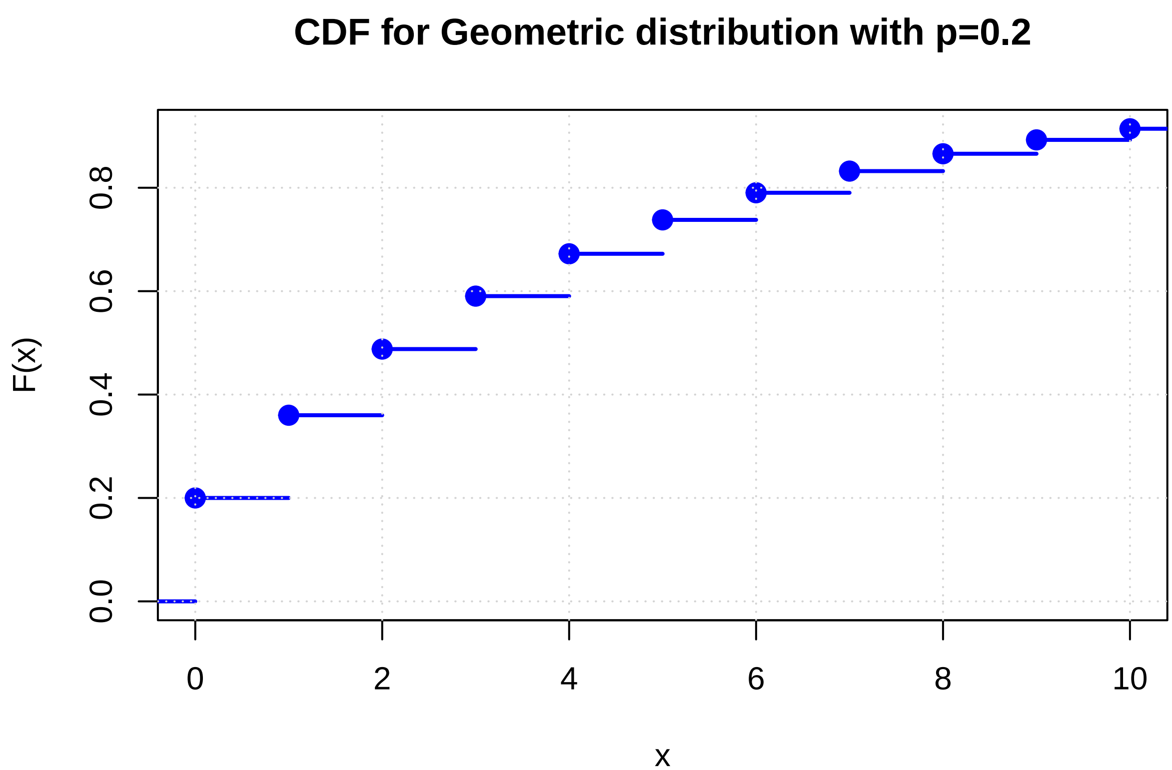
```
## min x such that P(X<=x)>=0.25
```

Plotting the pmf and cdf of Geometric distribution

```
p=0.2
xend=10
xx=0:xend
## Plot pmf
plot(xx,dgeom(xx,p),type="h",col="blue",lwd=2,main="PMF for Geomteric
     distribution with p=0.2",xlab="x",ylab="f(x)")
points(xx,dgeom(xx,p), pch=20, cex=2, col="blue")
```




```
## Plot CDF
yy=c(0,pgeom(xx,p))
plot(stepfun(xx,yy),vertical=F, pch=20, cex=2, lwd=2, col="blue",
     ylab="F(x)", main ="CDF for Geometric distribution with p=0.2", xlim=c(xx[1], xend))
grid()
```



Simulation from Geometric distribution

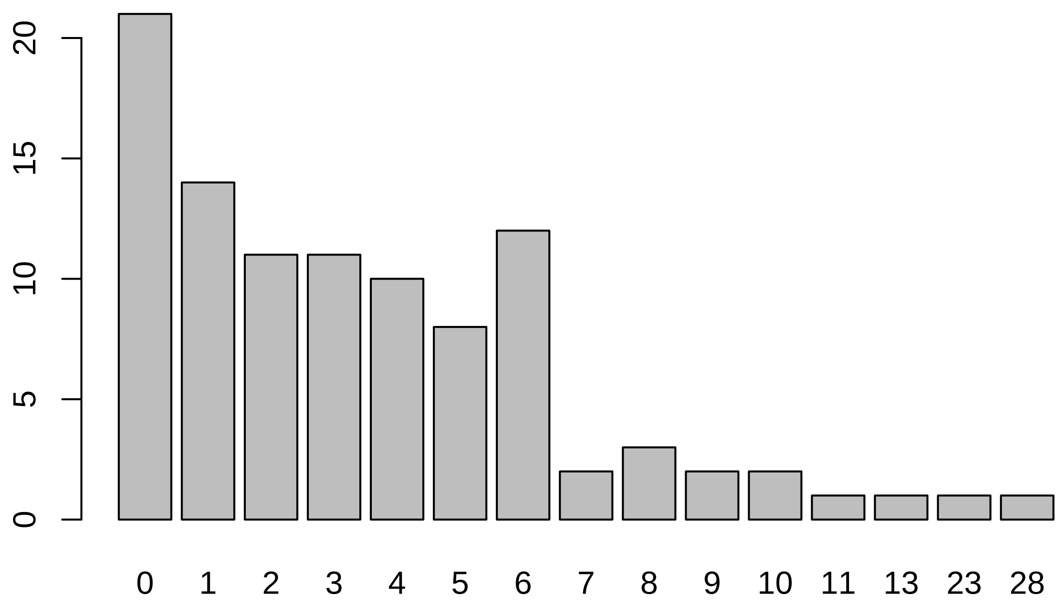
```
p=0.2
rgeom(10,p) # Geneartes 10 random samples from a Geometric distribution
```

```
##  [1]  0  1 10  0  5  4  2  3  4  5
```

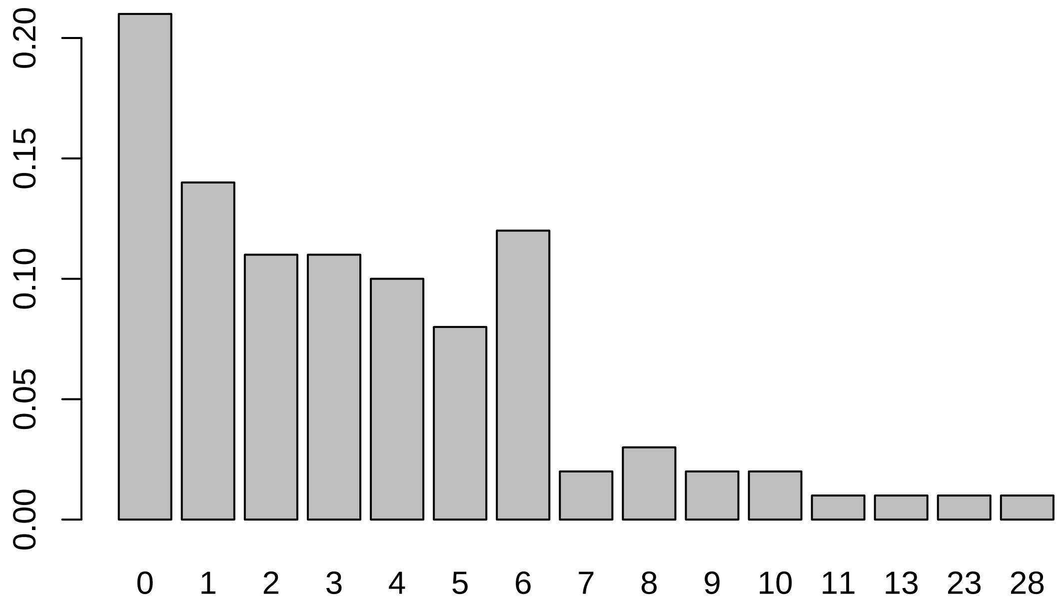
```
ns=100 ## sample size
set.seed(20) ## setting seed fixes the random number generator
x2= rgeom(ns,p)
table(x2) ## Frequency table for the sample
```

```
## x2
##  0  1  2  3  4  5  6  7  8  9 10 11 13 23 28
## 21 14 11 11 10  8 12  2  3  2  2  1  1  1  1
```

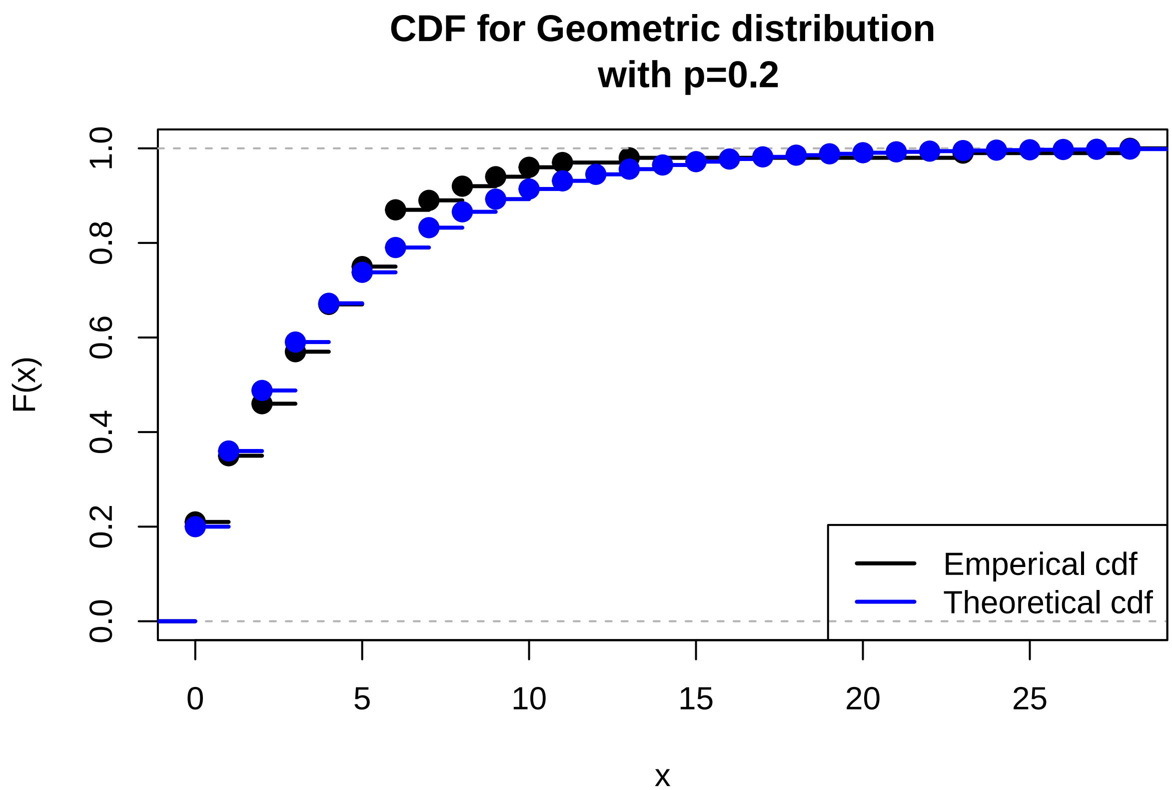
```
barplot(table(x2)) ## barplot for the frequency
```




```
barplot(prop.table(table(x2)) ) ## barplot using percentages
```




```
### Plotting emperical cdf of the simulated data and the theoretical cdf overlay
xx=0:max(x2)
yy=yy=c(0,pgeom(xx,p))
plot(ecdf(x2), lwd=2, pch=20, cex=2, main ="CDF for Geometric distribution
     with p=0.2", ylab="F(x)", xlim=c(xx[1], max(x2)))
lines(stepfun(xx,yy),vertical=F, pch=20, cex=2, lwd=2, col="blue")
legend("bottomright", c("Emperical cdf", "Theoretical cdf"), lty=c(1,1),
       lwd=c(2,2), col=c("black", "blue"))
```



Hypergeometric Distribution

Find probabilities related to this distribution Example: Let X be Hypergeometric random variable with N=20, K=8, n=5. N: Total number of balls (m+n in R) K: Number of white balls (m in R) n: Number of balls drawn without replacment (k in R)

```
?dhyper ## pmf for Hypergeometric distribution
n=5
N=20
K=8
dhyper(2,K,N-K,n) ## Finds P(X=2)  dhyper(x,m,n,k)
```

```
## [1] 0.3973168
```

```
phyper(2,K,N-K,n) ## Finds P(X <= 2) = P(X=0)+P(X=1)+P(X=2)
```

```
## [1] 0.7038184
```

```
mu = n*K/N ## mean
s2 = n*K/N*(N-K)/N*(N-n)/(N-1) ## variance
mu
```

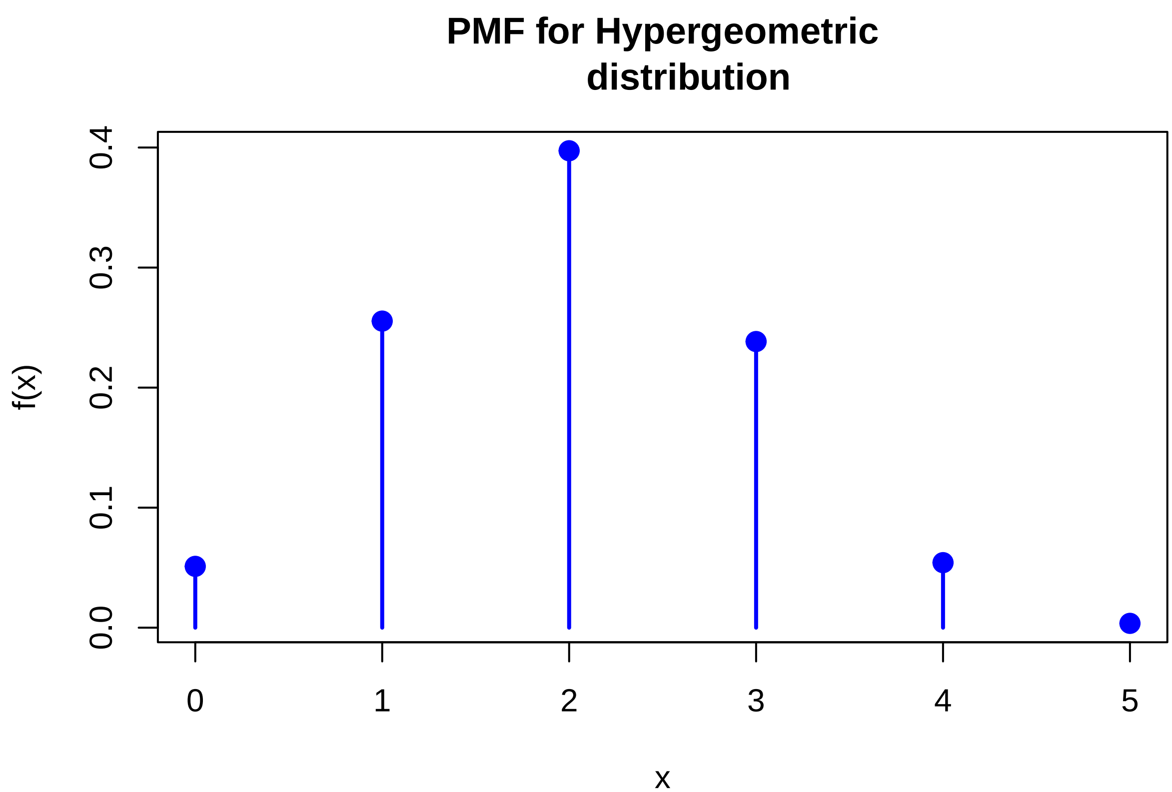
```
## [1] 2
```

```
s2
```

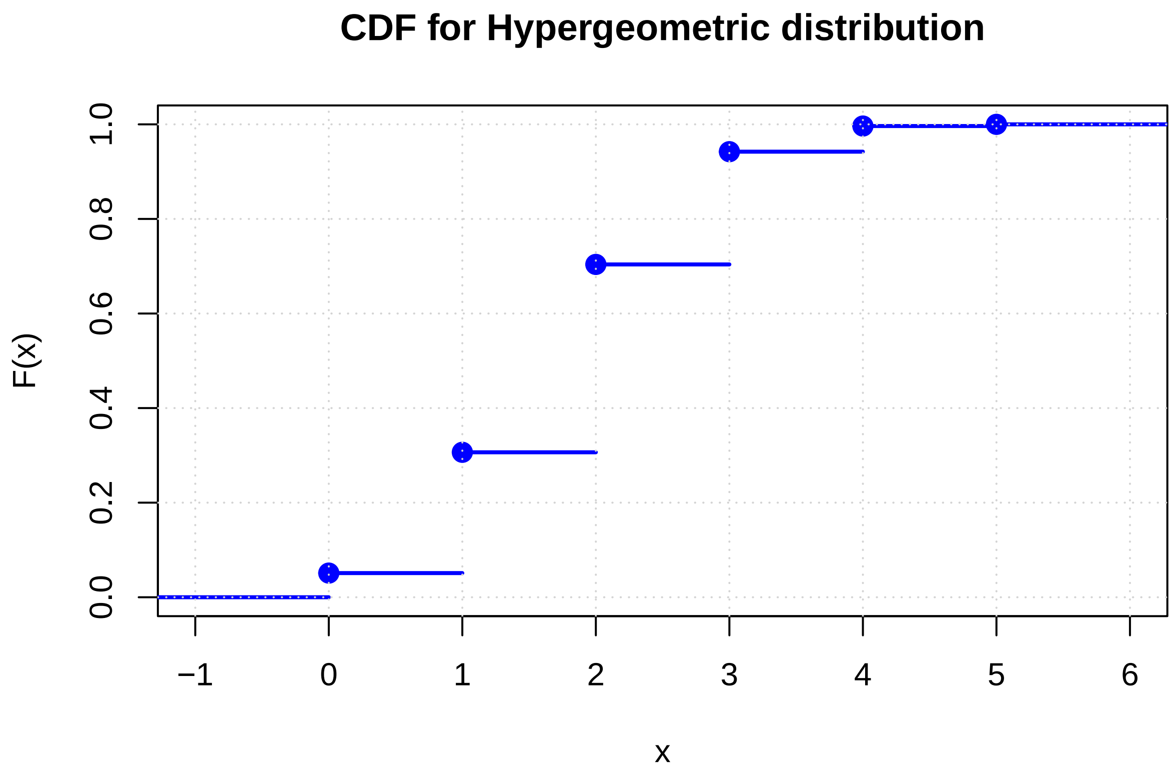
```
## [1] 0.9473684
```

Plotting the pmf and cdf of Hyergeometric distribution

```
xx=0:n
## Plotting pmf
plot(xx,dhyper(xx,K,N-K,n),type="h",col="blue",lwd=2,main="PMF for Hypergeometric
     distribution",xlab="x",ylab="f(x)")
points(xx,dhyper(xx,K,N-K,n), pch=20, cex=2, col="blue")
```




```
## Plotling CDF
yy=c(0,phyper(xx,K,N-K,n))
plot(stepfun(xx,yy),vertical=F, pch=20, cex=2, lwd=2, col="blue", ylab="F(x)",
     main ="CDF for Hypergeometric distribution")
grid()
```

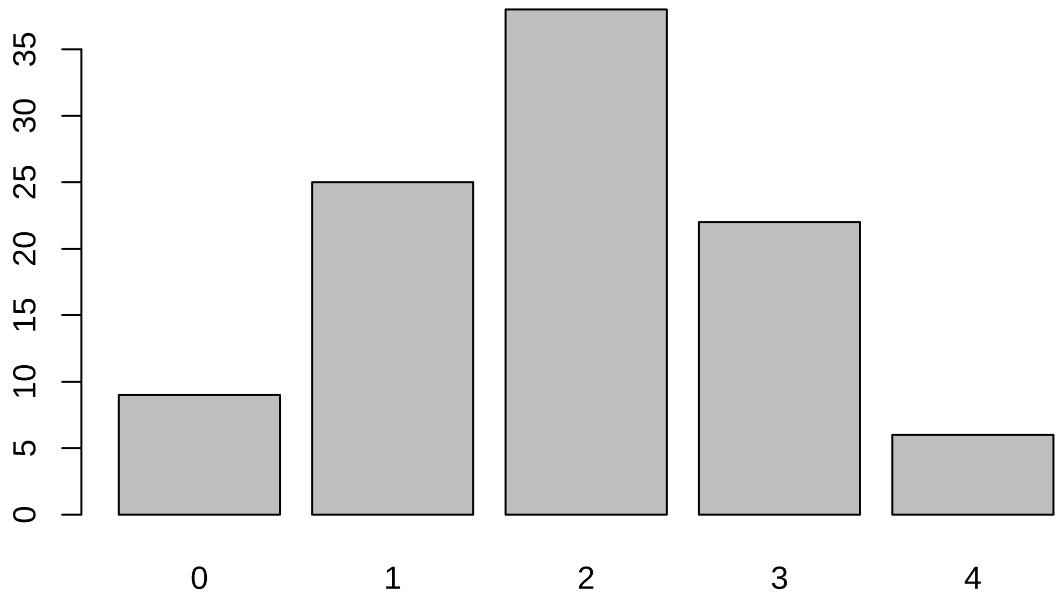


Simulation from Hypergeometric distribution

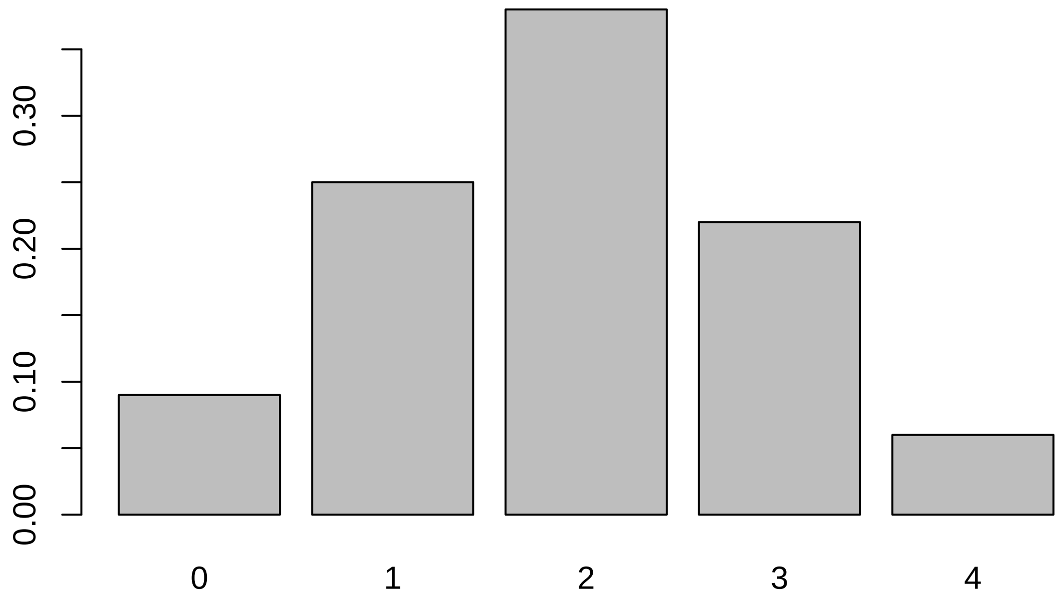
```
rhyper(10,K,N-K,n) # Geneartes 10 random samples from a Hypergeometric distribution
```

```
##  [1] 3 1 2 3 2 2 1 2 3 2
```

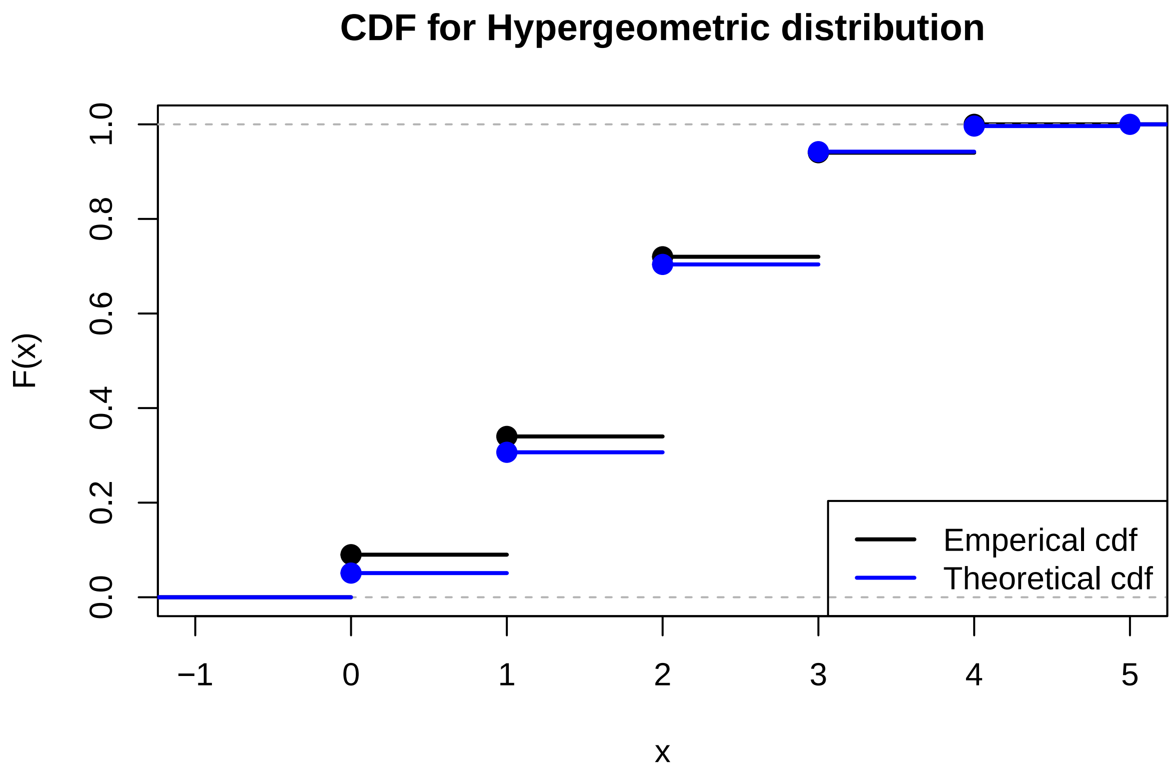
```
ns=100 ## sample size
set.seed(20) ## setting seed fixes the random number generator
x= rhyper(ns,K,N-K,n)
tx=table(x) ## Frequency table for the sample
barplot(table(x)) ## barplot for the frequency
```




```
barplot(prop.table(table(x))) ## barplot using probabilities
```




```
### Plotting emperical cdf of the simulated data and the theoretical cdf overlay
## ecdf (F_n(x)): (Number of x_i <=x)/ns
plot(ecdf(x), lwd=2, pch=20, cex=2, main ="CDF for Hypergeometric distribution",
     ylab="F(x)")
xx=0:n
yy=c(0,phyper(0:n,K,N-K,n))
lines(stepfun(0:n,yy),vertical=F, pch=20, cex=2, lwd=2, col="blue")
legend("bottomright", c("Emperical cdf", "Theoretical cdf"), lty=c(1,1),
       lwd=c(2,2), col=c("black", "blue"))
```



Quantile function for Hypergeometric distribution Quantile Definition: $Q(r) = inf\{x \in R : F(x) \geq r\}$

```
qhyper(0.5, K,N-K,n) ## Finds the value smallest x such that P(X<=x)>=0.1 (quantile)
```

```
## [1] 2
```

```
phyper(1,K,N-K,n)
```

```
## [1] 0.3065015
```

```
phyper(2,K,N-K,n)
```

```
## [1] 0.7038184
```

Poisson distribution

Example: X follows a Poisson distribution of lambda=2 (mean)

```
?dpois
l=2
dpois(3,l) ## P(X=3)
```

```
## [1] 0.180447
```

```
ppois(2,l) ## P(X<=2)
```

```
## [1] 0.6766764
```

```
rpois(10,l) ## Generate 10 random samples from Poisson distribution
```

```
##  [1] 4 5 2 7 2 2 4 3 3 4
```

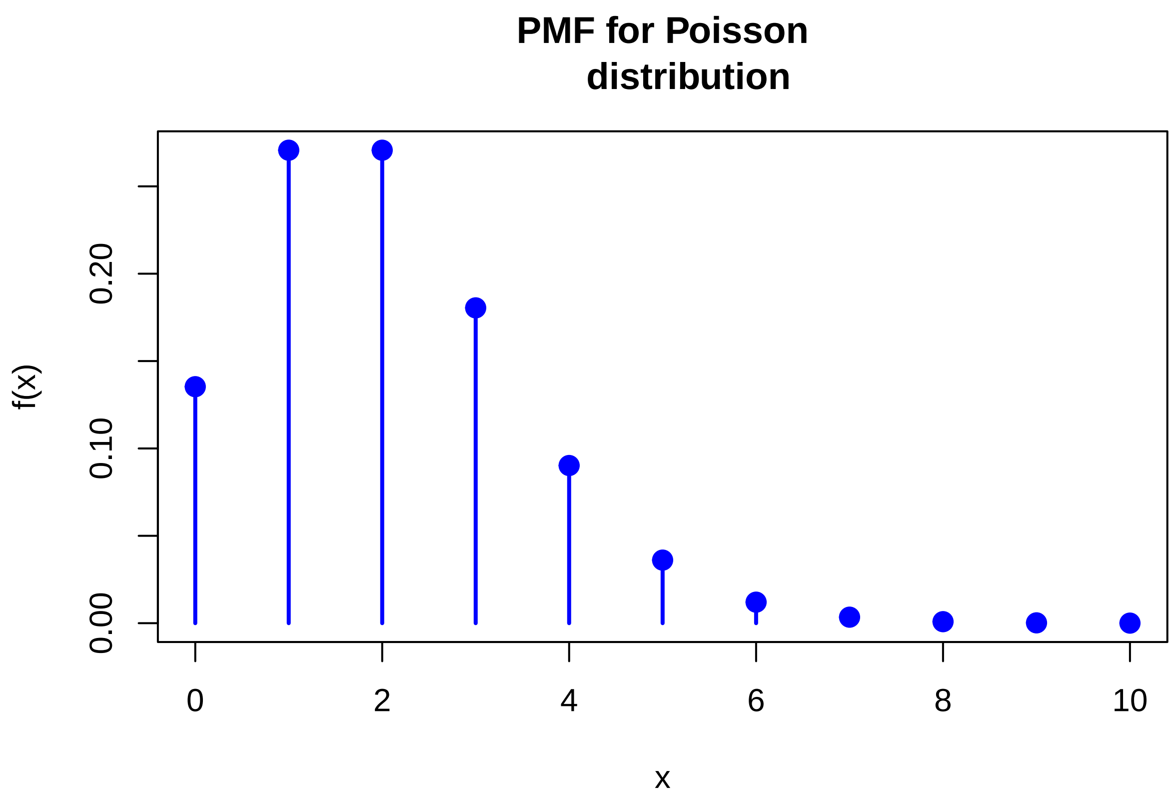
```
qpois(0.25, l) ## 25th quantile or 1st quartile of the Poisson distribution
```

```
## [1] 1
```

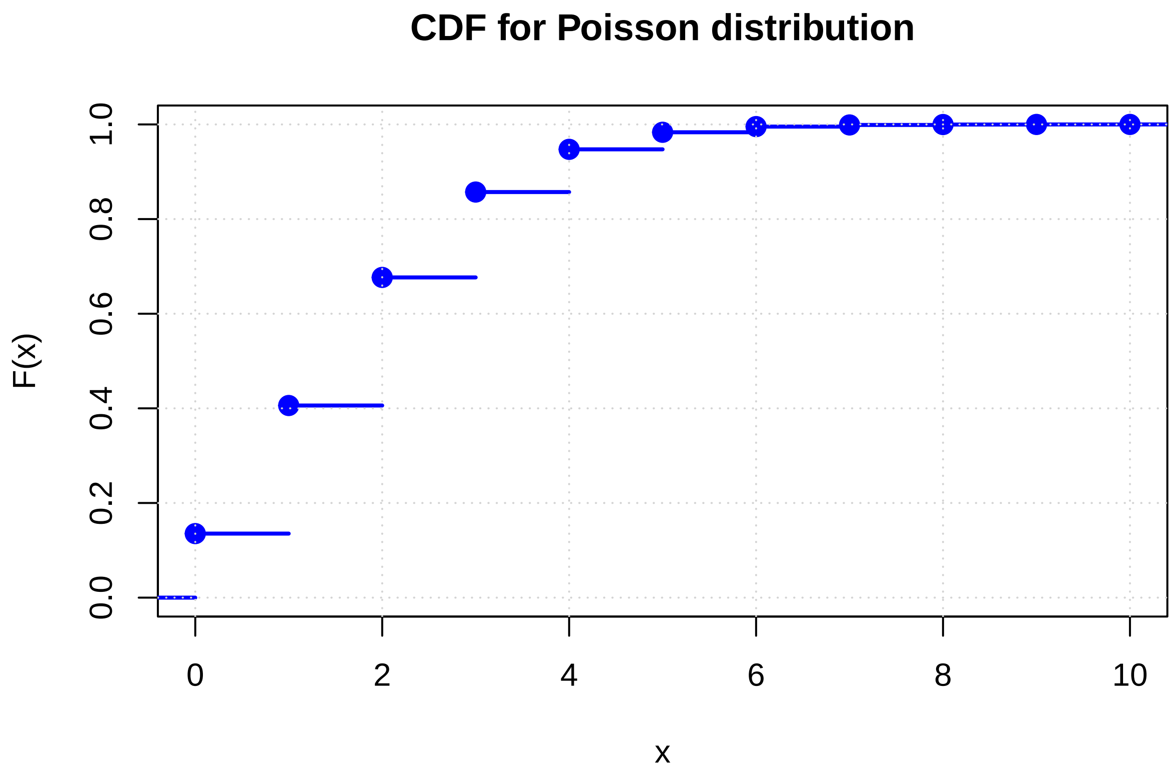
```
## min x such that P(X<=x)>=0.25
```

Plotting the pmf and cdf of Poisson distribution

```
l=2
xend=10
xx=0:xend
## Plot pmf
plot(xx,dpois(xx,l),type="h",col="blue",lwd=2,main="PMF for Poisson
     distribution",xlab="x",ylab="f(x)")
points(xx,dpois(xx,l), pch=20, cex=2, col="blue")
```




```
## Plot CDF
yy=c(0,ppois(xx,l))
plot(stepfun(xx,yy),vertical=F, pch=20, cex=2, lwd=2, col="blue",
     ylab="F(x)", main ="CDF for Poisson distribution",
     xlim=c(xx[1], xend))
grid()
```



Simulation from Poisson distribution

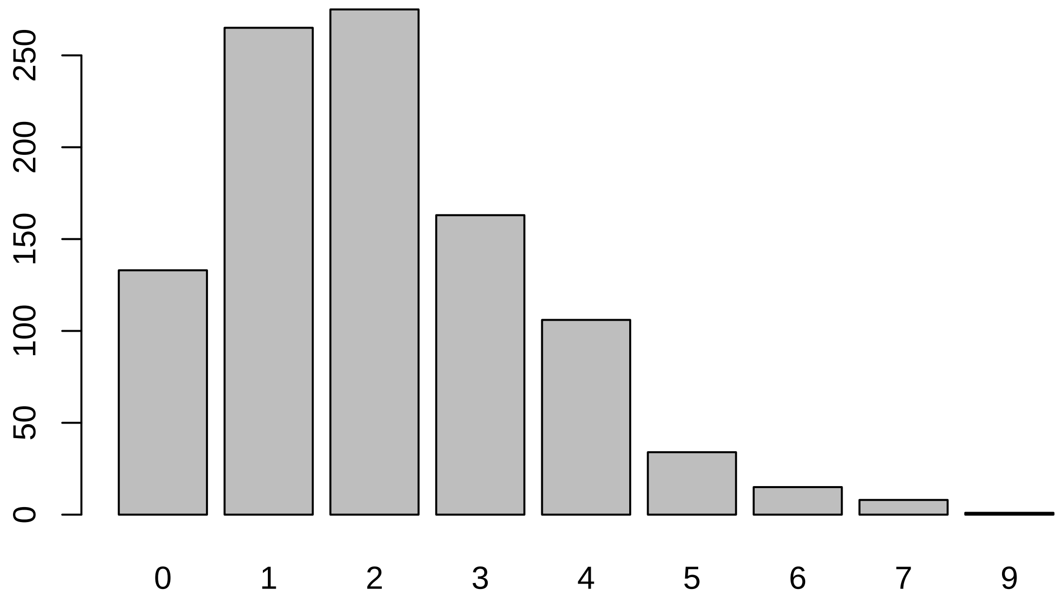
```
l=2
rpois(10,l) # Geneartes 10 random samples from a Poisson distribution
```

```
##  [1] 2 1 4 6 1 3 1 0 2 2
```

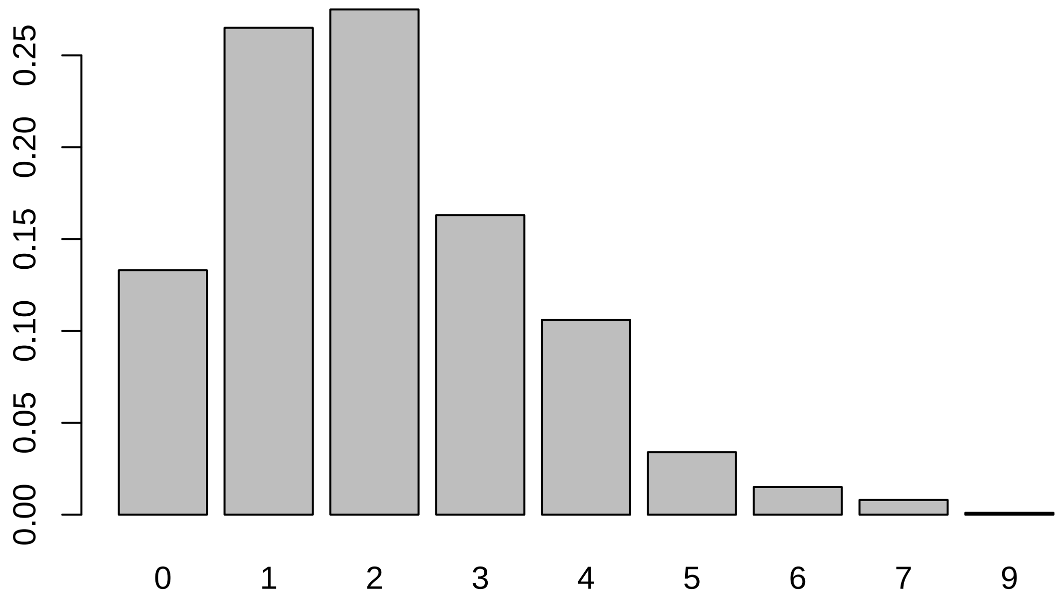
```
ns=1000 ## sample size
set.seed(20) ## setting seed fixes the random number generator
x2= rpois(ns,l)
table(x2) ## Frequency table for the sample
```

```
## x2
##   0   1   2   3   4   5   6   7   9
## 133 265 275 163 106  34  15   8   1
```

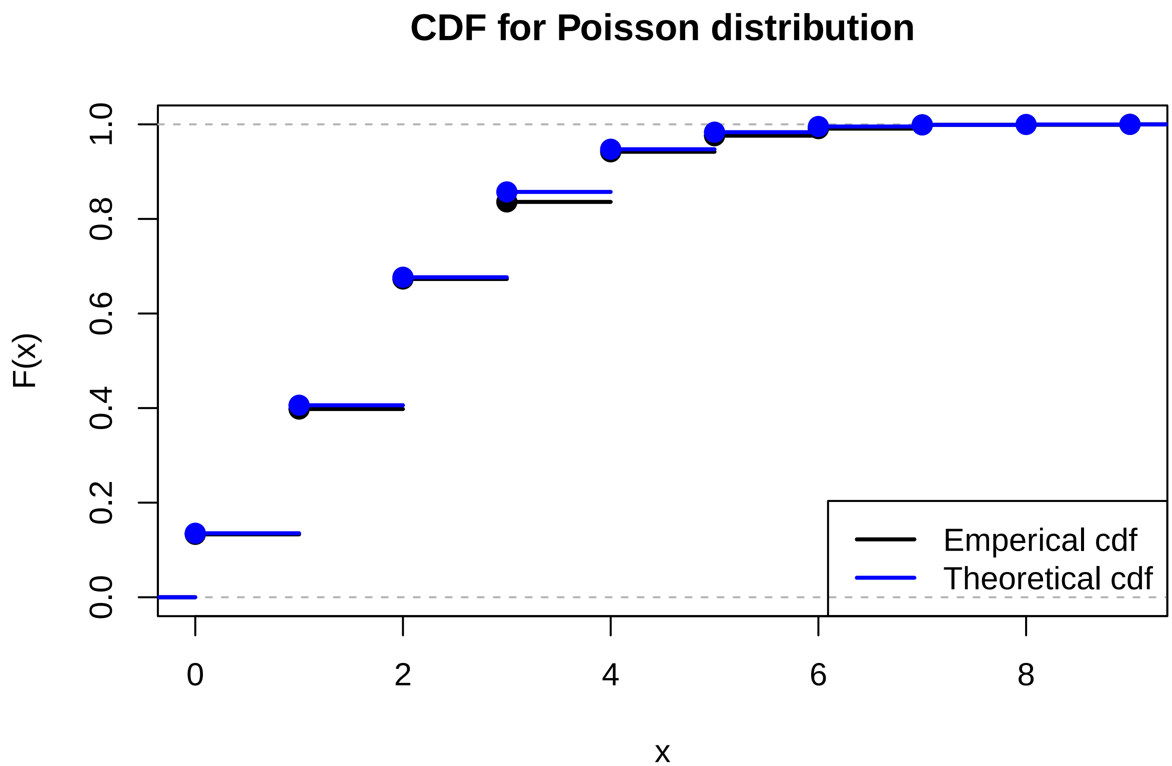
```
barplot(table(x2)) ## barplot for the frequency
```




```
barplot(prop.table(table(x2)) ) ## barplot using percentages
```




```
### Plotting emperical cdf of the simulated data and the theoretical cdf overlay
xx=0:max(x2)
yy=yy=c(0,ppois(xx,1))
plot(ecdf(x2), lwd=2, pch=20, cex=2, main ="CDF for Poisson distribution",
     ylab="F(x)", xlim=c(xx[1], max(x2)))
lines(stepfun(xx,yy),vertical=F, pch=20, cex=2, lwd=2, col="blue")
legend("bottomright", c("Emperical cdf", "Theoretical cdf"), lty=c(1,1),
       lwd=c(2,2), col=c("black", "blue"))
```


CDF for Poisson distribution
F(x)
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
x
Emperical cdf
Theoretical cdf