Exercise 5 - Selected distributions of a discrete random variable

Martina Litschmannová, Adéla Vrtková, Michal Béreš

Overview of distributions and their functions

Introduction: Probability, Cumulative Probability(Distribution) and Quantile functions

Probability function

- starts with the letter **d**: $p = P(X = x)$: p=d...(x,...)

Cumulative Probability(Distribution Function)

- starts with the letter **p**: $p = P(X \leq x)$:p=p...(x,...)
- note that Cumulative probability in R is with the alternative definition $P(X \leq t)$
- for our distribution function $F(t) = P(X < t)$: F(t)=p...(t - 1,...)

Quantile function

- starts with the letter **q**: $p \geq P(X \leq x)$: x=q...(p,...)
- searches for the smallest x for which $P(X \leq x)$ is greater than p

Binomial(Alternative): $X \sim Bi(n, \pi), X \sim A(\pi) = Bi(1, \pi)$

- number of successes in n Bernoulli attempts(or for one attempt in the case of Alternative dist.)
- every attempt has a chance of success π

In [1]:
```
# Probability function P(X=x)
x = 10    # value for which we are looking for a p-st function
n_count = 21    # selection range
p_prob = 0.5  # probability of success
dbinom(x, size=n_count,prob=p_prob)
```

0.168188095092773

In [2]:
```
n = 21    # selection range
p = 0.5   # probability of success
```

In [3]:
```
dbinom(3.2, n, p)
```

```
Warning message in dbinom(3.2, n, p):
“non-integer x = 3.200000”
```

0

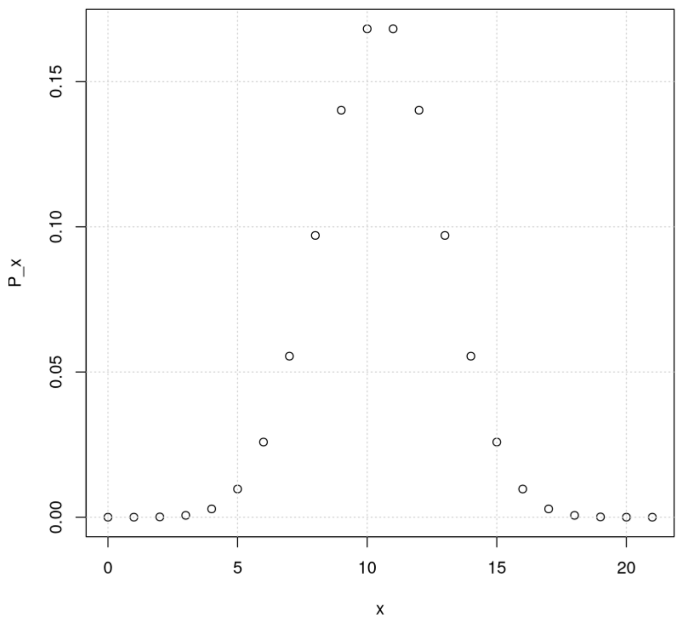
In [4]:
```
options(warn=-1) # this can be used to turn off warnings
```

In [5]:
```
dbinom(3.2, n, p)
```

0

In [6]:
```
options(warn=0) # this will switch it on again
```

In [7]:
```
# draw a probability function
x = 0:21 # minimum 0, maximum n has a positive probability
P_x = dbinom(x, n, p)
plot(x, P_x)
grid()
```



In [8]:
```
# Cumulative probability function P(X<=x)
x = 10   # value for which we are looking for a value of cumulative prob function
n = 21   # selection range
p = 0.5  # probability of success
pbinom(x, n, p)
```

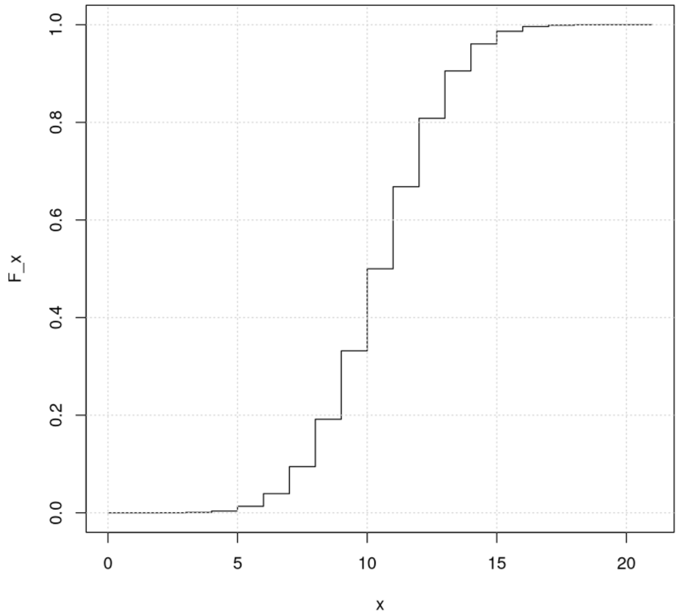
0.5

In [9]:
```
# Distribution function F(x)=P(X<x)
x = 10   # value for which we are looking for a value of cumulative prob function
n = 21   # selection range
p = 0.5  # probability of success
pbinom(x, n, p) - dbinom(x, n, p)
# or
pbinom(x - 1, n, p)
```

0.331811904907226

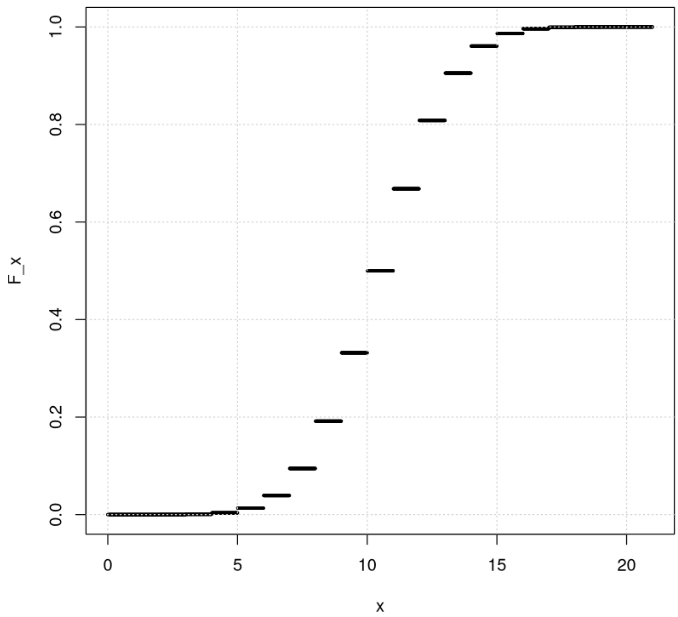
0.331811904907227

In [10]:
```
# we plot the distribution function
x = 0:21 # minimum 0, maximum n has a positive probability
P_x = dbinom(x, n, p)
F_x = cumsum(P_x)
plot(x, F_x, type='s')
grid()
```



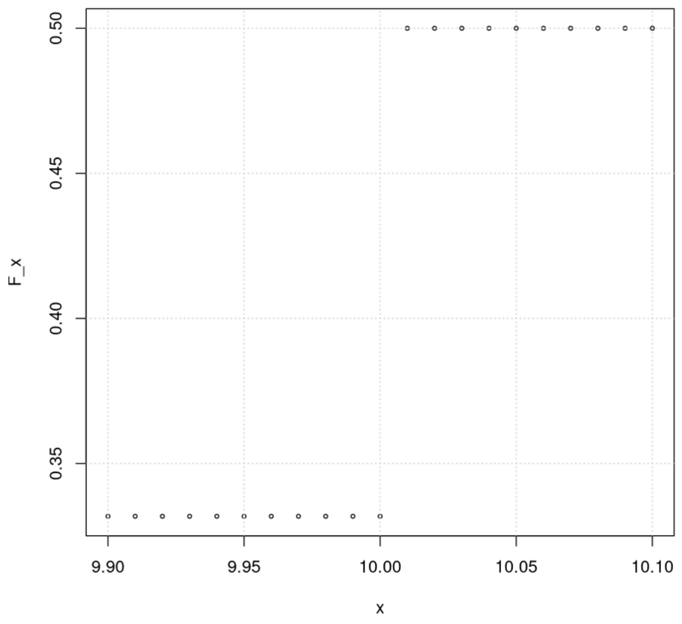
In [11]:

```
# or
x = seq(0, 21, 0.01) # minimum 0, maximum n
options(warn=-1)
F_x = pbinom(x, n, p) - dbinom(x, n, p)
plot(x, F_x, cex=0.3)
grid()
options(warn=0)
```



In [12]:

```
# check correctness at 10
x = seq(9.9, 10.1, 0.01) # minimum 0, maximum n
options(warn=-1)
F_x = pbinom(x, n, p) - dbinom(x, n, p)
options(warn=0)
plot(x, F_x, cex=0.5)
grid()
```



In [13]:

```
# find x for given q: q=P(X<=x)
q = 0.7  # h
n = 21   # selection range
p = 0.5  # probability of success
qbinom(q, n, p)
pbinom(11, n, p)
pbinom(12, n, p)
pbinom(13, n, p)
```

12

0.668188095092773

0.808344841003418

0.905376434326172

In [14]:

```
# Quantile function(inversion of dist. Function): q=F(x)=P(X<x)
q = 0.7   # probability for which we are looking for a quantile
n = 21   # selection range
p = 0.5  # probability of success
qbinom(q, n, p) + 1
pbinom(12 - 1, n, p)
pbinom(13 - 1, n, p)
pbinom(14 - 1, n, p)
```

13

0.668188095092773

0.808344841003418

0.905376434326172

Hypergeometric: $X \sim H(N, M, n)$

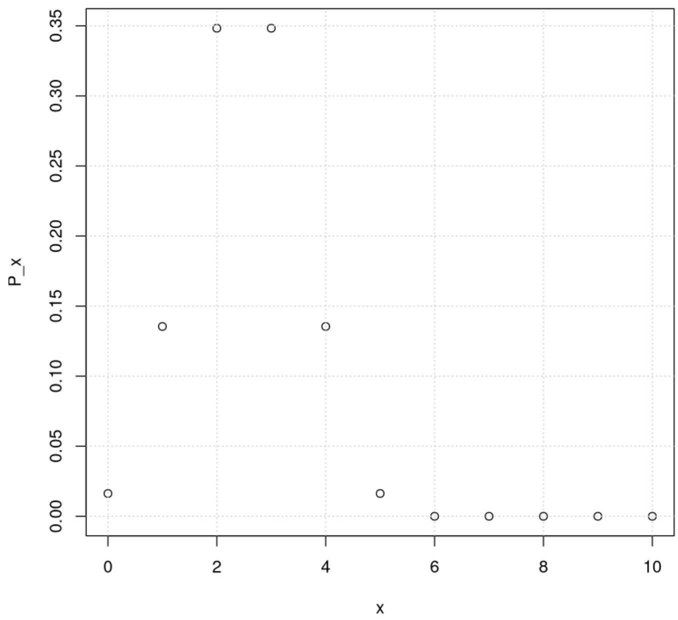
- number of successes in n dependent attempts
- type dependence:
  - N objects,
  - of which M objects with specified property,
  - size of the selection n
  - **we do not return back when selecting - the probability of selecting an object with a given property changes with each additional selected object**

- **R function takes as parameters * hyper(k, M, N - M, n)**
- k is the number of successes for which we calculate the probability,
- M is the number of objects with the specified property,
- NM is the number of objects without the specified property,
- n is the target size of the selection.

In [15]:
```
# Probability function P(X=x)
x = 5   # value for which we are looking for a p-st function
N = 20  # total number of objects
M = 5   # of which with specified property
n = 10  # selection size
dhyper(x, M, N - M, n)
```

0.0162538699690403

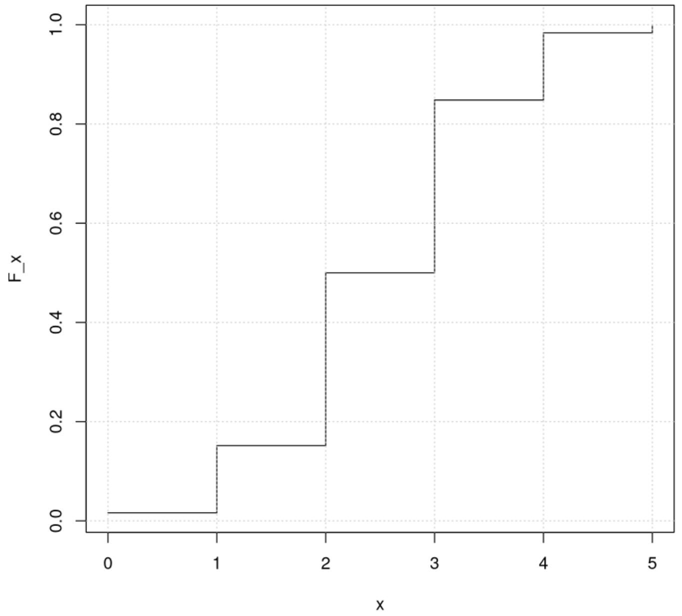
In [16]:
```
# plot a probability function
x = 0:10 # minimum 0, maximum n or M has a positive truth.
P_x = dhyper(x, M, N - M, n)
plot(x, P_x)
grid()
```



In [17]:
```
# Distribution function F(x)=P(X<x)
x = 5   # value for which we are looking for dist. function
N = 20  # total number of objects
M = 5   # of which with specified property
n = 10  # selection size
phyper(x - 1, M, N - M, n)
```

0.98374613003096

In [18]:
```
# plot the Distribution function
x = 0:5 # minimum 0, maximum n or M has a positive truth.
P_x = dhyper(x, M, N - M, n)
F_x = cumsum(P_x)
plot(x, F_x, type='s')
grid()
```



In [19]:

```
# Quantile function(inversion of dist. Function): q=P(X<x)
q = 0.7 # probability for which we are looking for a quantile
N = 20  # total number of objects
M = 5   # of which with specified property
n = 10  # selection size
qhyper(q, M, N - M, n) + 1
phyper(3 - 1, M, N - M, n)
phyper(4 - 1, M, N - M, n)
phyper(5 - 1, M, N - M, n)
```

4

0.5

0.848297213622291

0.98374613003096

Negative binomial(Geometric): $X \sim NB(k, \pi), X \sim Ge(\pi) = NB(1, \pi)$

- number of attempts up to k-th success(inclusive)
- Every attempt has a chance of success π
- **Negatively binomial NV is defined in R as the number of failures before the success**
- therefore we will send x - k as the first parameter

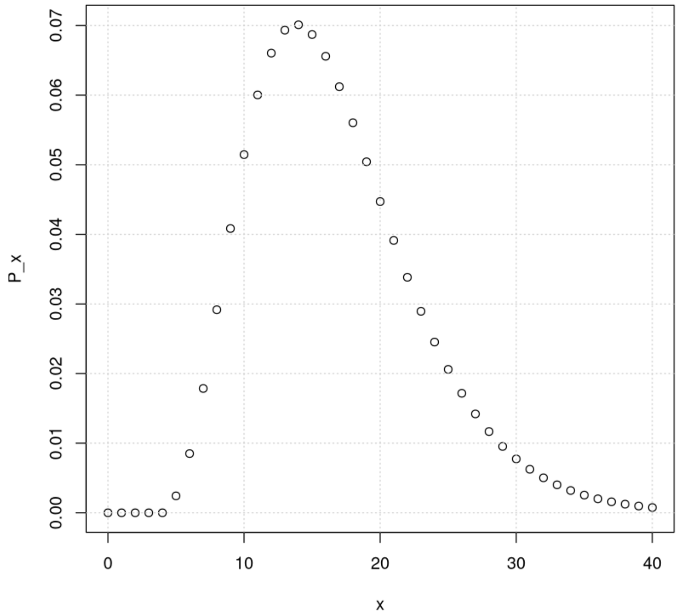
In [20]:

```
# Probability function P(X=x)
x = 15   # number of attempts for which we are looking for prob. fc
k = 5    # required number of successes
p = 0.3  # prob. of individual trials
# Note that the first argument must be the number of failures
dnbinom(x - k, k, p)
```

0.068710126992507

In [21]:

```
# let's plot the probability function
x = 0:40 # minimum k, maximum unlimited
P_x = dnbinom(x - k, k, p)
plot(x, P_x)
grid()
# values 0,1,2,3,4 have P(x)=0
```



In [22]:

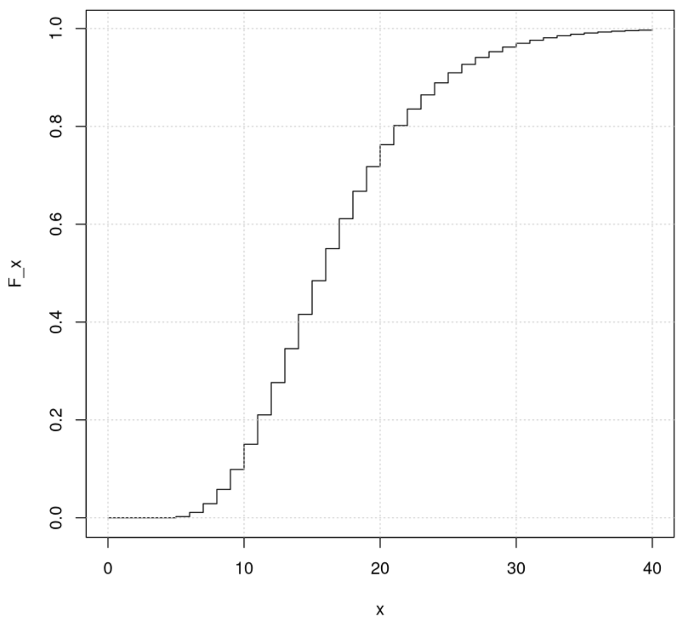
```
# Distribution function F(x)=P(X<x)
x = 15   # number of attempts for which we are looking for prob. fci
k = 5    # required number of successes
p = 0.3  # true individual trials
# Note that the first argument must be the number of failures
pnbinom(x - k - 1, k, p)
pnbinom(x - k, k, p)-dnbinom(x - k, k, p)
```

0.41579881378065

0.41579881378065

In [23]:

```
# let's draw the Distribution function
x = 0:40 # minimum 0, maximum n or M has a positive prob.
P_x = dnbinom(x - k, k, p)
F_x = cumsum(P_x)
plot(x, F_x, type='s')
grid()
```



In [24]:

```
# Quantile function(inversion of dist. Function): q=P(X<x)
q = 0.7  # prob. for quantile
k = 5    # required number of successes
```

```
p = 0.3  # true individual trials
qnbinom(q, k, p) + k + 1
pnbinom(19 - k - 1, k, p)
pnbinom(20 - k - 1, k, p)
pnbinom(21 - k - 1, k, p)
```

20

0.667345147229023

0.717776458479928

0.762492221122399

Poisson: $X \sim Po(\lambda t)$

- number of events in the Poisson process in a closed area(in time, area, volume)
- with occurrence density λ
- in time/area/volume of size t

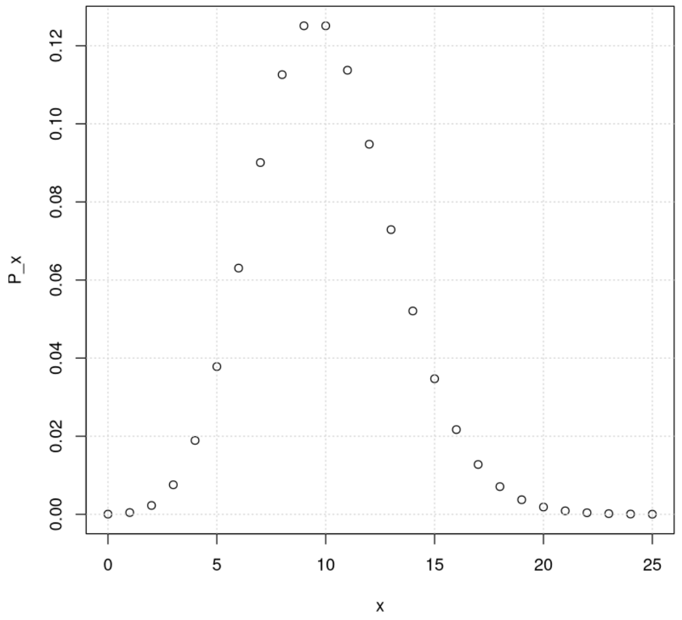
In [25]:

```
# Probability function P(X=x)
x = 9        # number of attempts for which we are looking for prob. fci
lambda = 5   # density of occurrence
t = 2        # area/time/volume
lt = lambda*t
dpois(x, lt)
```

0.125110035721133

In [26]:

```
# draw a probability function
x = 0:25 # minimum 0, maximum unlimited
P_x = dpois(x, lt)
plot(x, P_x)
grid()
```



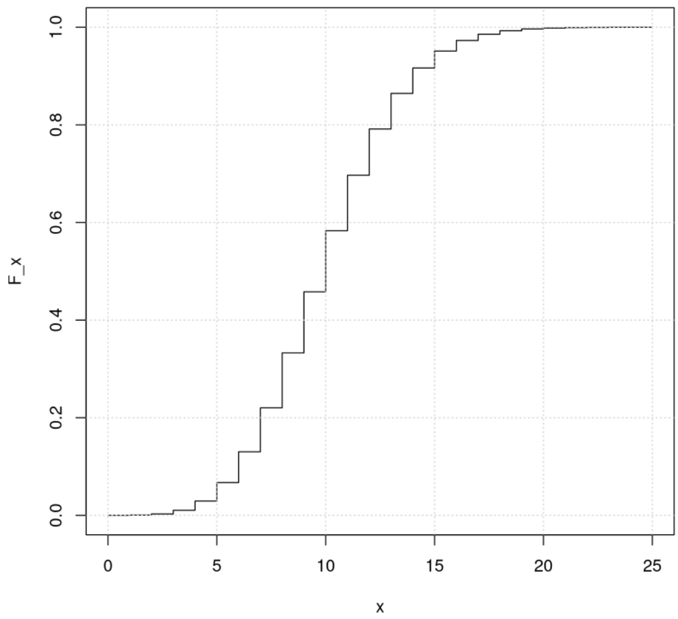
In [27]:

```
# Distribution function F(x)=P(X<x)
x = 9        # number of attempts for which we are looking for prob. fci
lambda = 5   # density of occurrence
t = 2        # time/area
lt = lambda*t
ppois(x - 1, lt)
```

0.332819678750719

In [28]:

```
# let's plot the Distribution function
x = 0:25 # minimum 0, maximum n or M has a positive prob.
P_x = dpois(x, lt)
F_x = cumsum(P_x)
plot(x, F_x, type='s')
grid()
```



In [29]:

```
# Quantile function(inversion of dist. Function): q=P(X<x)
q = 0.4      # prob. for quantile
lambda = 5   # density of occurrence
t = 2        # time/area
lt = lambda*t
qpois(q, lt) + 1
ppois(9 - 1, lt)
ppois(10 - 1, lt)
ppois(11 - 1, lt)
```

10

0.332819678750719

0.457929714471852

0.583039750192985

Examples

Example 1.

Bridge is played with 52 bridge cards, which are dealt among 4 players. There are always 2 players playing together. When dealing(13 cards) you received 2 aces. What is the probability that your partner will have the remaining two aces?

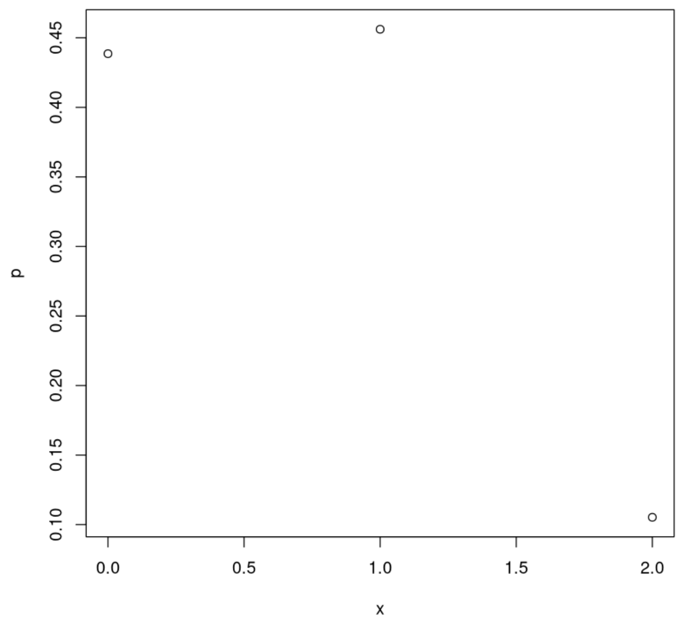
In [30]:

```
# X... number of aces among 13 cards
# X~H(N=39, M=2, n=13)
# P(X=2)
M = 2
N = 39 # 52-13
n = 13
# calculation
dhyper(2, M, N - M, n) # which is dhyper(2,2,37,13)
```

0.105263157894737

In [31]:

```
# probability function graph
x = 0:M    # all possible implementations of NV X.
p = dhyper(x, M, N - M, n) # values of the probability function for x
plot(x, p)
```



Example 2.

Experiments have shown that a radioactive substance emits within 7.5 s an average of 3.87 α-particles. Determine the probability that this substance will emit at least one α-particle in 1 second.

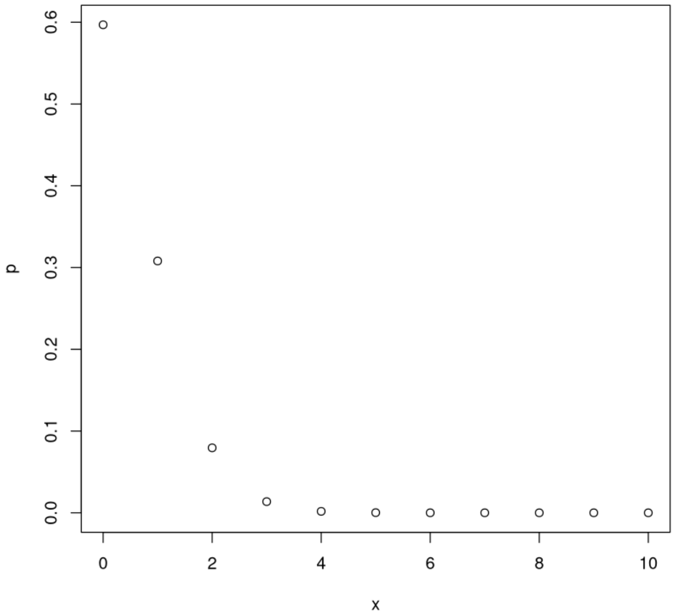
In [32]:

```
# X... number of radiated alpha particles during 1 s
# X~Po(lt=3.87/7.5)

lambda = 3.87/7.5 # frequency of occurrence
t = 1 # in 1 second
lt = lambda*t # Poisson distribution parameter

# P(X>=1)=P(X>0)=1 - P(X<=0)
1 - ppois(1 - 1, lt)
```

0.403096607325642

In [33]:

```
# probability function graph
# theoretically up to an infinite number of particles can be emitted,
# from a certain value the probability is negligible
x = 0:10
p = dpois(x, lt) # probability function values for x
plot(x, p)
```



Example 3.

A friend sends you to the cellar to bring 4 bottled beers - two 10˚ and two 12˚. You don't know where to turn the light on, so you take 4 bottles blindly. How likely were you to comply with the requirement if you knew that there were a total of 10 10˚ and 6 12˚ in the base?

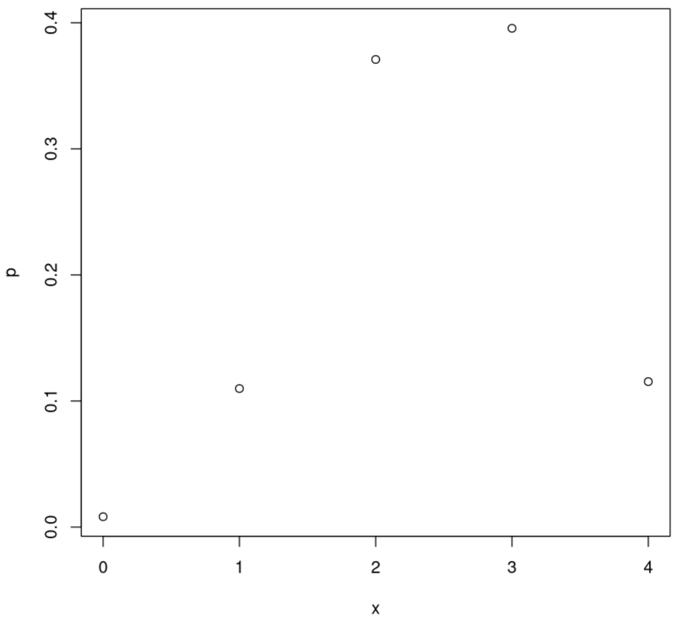
In [34]:

```
# X... number of 10 ° beers among 4 selected
# X~H(N=16, M=10, n=4)

x = 2
N = 16
M = 10
n = 4

# P(X=2)
dhyper(x, M, N - M, n)
```

0.370879120879121

In [35]:

```
# graph of probability function
x = 0:4    # all possible implementations of NV X.
p = dhyper(x, M, N - M, n) # values of the probability function for x
plot(x, p)
```



Example 4.

On average, there are 15 certain microorganisms in one milliliter of a perfectly mixed solution. Determine the probability that there will be less than 5 of these micro-organisms in a test tube if a sample of 1/2 milliliter is randomly selected.

In [36]:

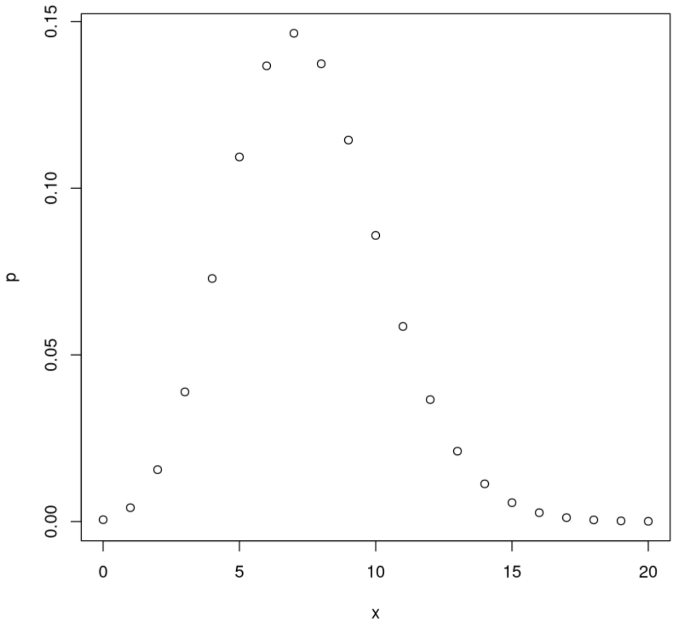
```
# X... number of microorganisms in 0.5 ml solution
# X~Po(lt=15/2)

lambda = 15
t = 1/2
lt = lambda*t   # Poisson distribution parameter

# P(X<5)=P(X<=4)
ppois(5 - 1, lt)
# or
ppois(5,lt) - dpois(5,lt)
```

0.132061856287721

0.132061856287721

In [37]:

```
# graph of probability function
# in theory there may be an infinite number of microorganisms in solution,
# from a certain value the probability is negligible
x = 0:20
p = dpois(x, lt) # values of the probability function for x
plot(x, p)
```



Example 5.

Throw 15 coins on the table. What is the probability that the number of coins lying face up is from 8 to 15?

In [38]:

```
# X... number of coins that fall face up out of a total of 15 coins
# X~Bi(n=15, p=0.5)

n = 15
p = 0.5

# P(8<=X<=15)=P(X<=15) - P(X<8)=P(X<=15) - P(X<=7)
pbinom(15, n, p) - pbinom(7, n, p)
```

0.5

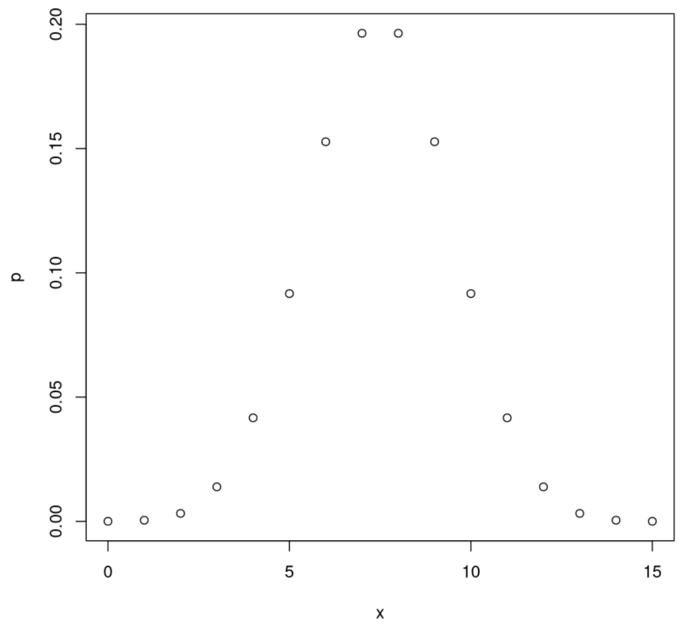
In [39]:

```
# otherwise: P(8<=X<=15)=P(X>7)=1-P(X<=7)
1 - pbinom(7, n, p)
```

0.5

In [40]:

```
# graph of probability function
x = 0:15    # all possible implementations of NV X.
p = dbinom(x, n, p) # values of the probability function for x
plot(x, p)
```



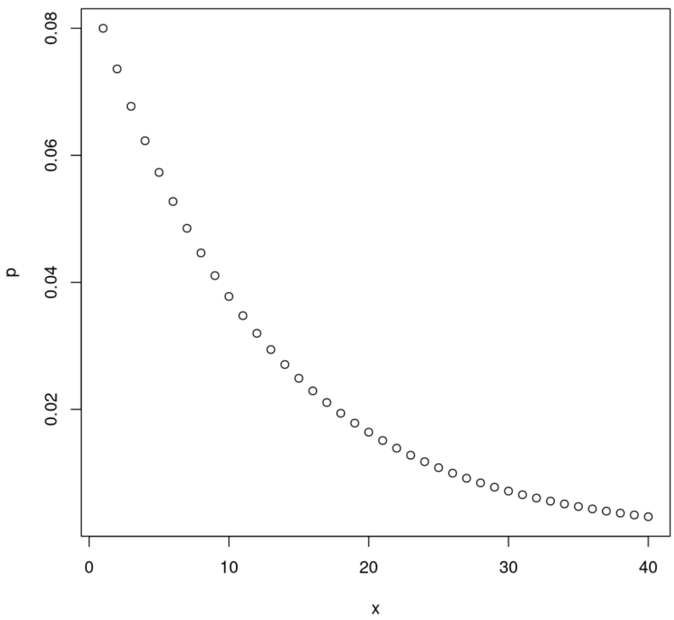
Example 6.

The probability that we will call the studio of the radio station that has just announced a telephone competition is 0.08. What is the probability that we will manage to get in on the 4th attempt at the most?

In [41]:
```
# X... number of attempts before we call the radio studio
# X~NB(k=1, p=0.08) or G(0.08)

x = 4
k = 1
p = 0.08

# P(X<=4)
pnbinom(x - k, k, p)
```

0.28360704

In [42]:
```
# graph of probability function
# theoretically we can make infinitely many attempts,
# from a certain value the probability is negligible
x = 1:40
p = dnbinom(x - k, k, p) # probability function values for x
plot(x, p)
```



Example 7.

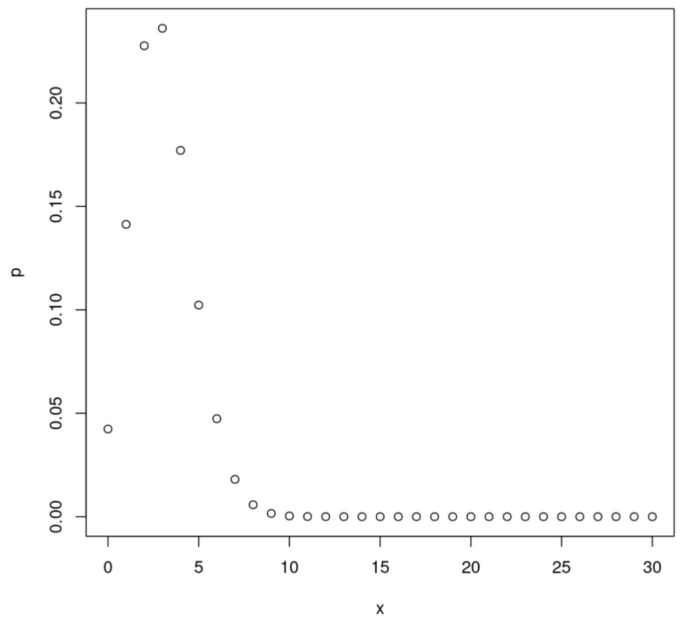
In average 10% of the components produce at the factory are defective. What is the probability that if we select thirty components from the daily production, at least two will be defective?

In [43]:
```
# X... number of defective parts out of 30 selected
# X~Bi(n=30, p=0.1)

n = 30
p = 0.1

# P(X>=2)=1 - P(X<2)=1 - P(X<=1)
1 - pbinom(1, n, p)

# or P(X>=2) all except 0 and 1
1 - (dbinom(0, n, p) + dbinom(1, n, p))
```

0.816304980807397

0.816304980807396

In [44]:
```
# graph of probability function
x = 0:30            # all possible implementations of NV X.
p = dbinom(x, n, p) # values of the probability function for x
plot(x,p)
```



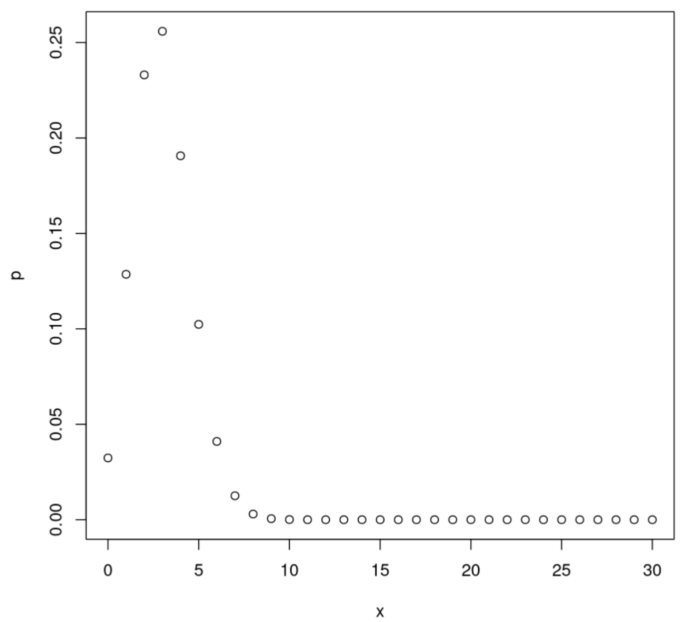
Example 8.

There are 200 parts in stock. 10% of them are defective. What is the probability that if we select thirty parts from the warehouse, at least two will be defective?

In [45]:
```
# X... number of defective parts out of 30 selected from 200
# X~H(N=200, M=20, n=30)

N = 200
M = 20
n = 30

# P(X>=2)=1 - P(X<2)=1 - P(X<=1)
1 - phyper(2 - 1, M, N - M, n)
```

0.839070983048613

In [46]:
```
# graph of probability function
x = 0:30    # all possible implementations of NV X.
p = dhyper(x, M, N - M, n) # probability function values for x
plot(x, p)
```



Example 9.

In a company, it was found that some illegal software was installed on 33% of computers. Determine the probability and distribution function of the number of computers with illegal software among the three computers inspected.

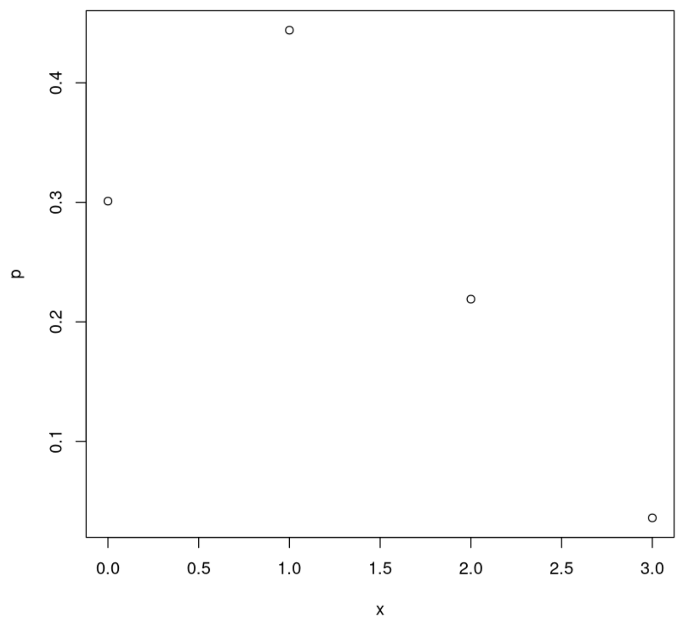
In [47]:
```
# X... number of computers with illegal software out of 3 checked
# X~Bi(n=3, p=0.33)

n = 3
p = 0.33

# probabilistic function
x = 0:3  # all possible implementations of NV X
p = dbinom(x, n, p) # probability function values for x

p = round(p, 3) # rounding probabilities to 3 des. places
p[4] = 1 - sum(p[1:3]) # completion of the last value by 1

tab = rbind(x, p) # Create probability function table
rownames(tab) = c("x", "P(x)")
tab
```

A matrix: 2 × 4 of type dbl

x	0.000	1.000	2.000	3.000
P(x)	0.301	0.444	0.219	0.036

In [48]:
```
# probability function graph
plot(x, p)

# distribution function
cumsum(p) # simplified distribution function listing
```

0.301 · 0.745 · 0.964 · 1



Example 10.

Sportka is a lottery game in which the bettor bets six numbers out of forty-nine, which he expects to fall in a future draw. To participate in the game, it is necessary to choose at least one combination of 6 numbers (always 6 numbers per column of 49 numbers) and use crosses to mark picked numbers. You can bet multiple times by filling multiple columns. The bettor wins if he guesses at least three numbers from the drawn six numbers. What is the probability that in order for the bettor to win, he will have to fill in:

In [49]:

```
# First the probability that we get in one column

# Y... number of guessed numbers in 6 drawn from 49
# Y~H(N=49, M=6, n=6)

N = 49
M = 6
n = 6

# P-st guess at least 3 numbers in one column
# P(Y>=3)=1 - P(Y<3)=1 - P(Y<=2)
pp = 1 - phyper(3 - 1, M, N - M, n)
pp
```

0.0186375450020223

a)

just three columns,

In [50]:
```
# X… the number of columns a bettor will have to fill in order to win
# X~NB(k=1, p=pp)

# a) P(X=3)
k = 1
p = pp

dnbinom(3 - k, k, pp)
```

0.0179493027365343

b)

at least 5 columns,

In [51]:
```
# b) P(X>=5)=1 - P(X<5)=1 - P(X<=4)

1 - pnbinom(5 - k - 1, k, pp)
```

0.927508193544096

c)

less than 10 columns,

In [52]:
```
# c) P(X<10)=P(X<=9)
pnbinom(10 - k - 1, k, pp)
```

0.155761898754964

d)

more than 5 and at most 10 columns?

In [53]:
```
# P(5<X<=10)=P(X<=10) - P(X<=5)
pnbinom(10 - k, k, pp)-pnbinom(5 - k, k, pp)
# or P(X<11) - P(X<6)
pnbinom(11 - k - 1, k, pp)-pnbinom(6 - k - 1, k, pp)
```

0.0817181422065138

0.0817181422065138

Example 11.

The probability of throwing 6 on a 6-wall cube is 1/6. We roll until we roll six 10 times.

a)

What is the mean value of the number of throws.

In [54]:
```
# X… rolls the dice before we roll 10 sixes
# X~NB(k=10, p=1/6)

k = 10
p = 1/6

E_X = k/p
E_X
```

60

b)

How many throws do we have to count on if we want the probability of throwing 10 sixes to be at least 70%.

In [55]:
```
# P(X<=k)>=0.7
qnbinom(0.7, k, p) + k
```

68