# Problema 1

## *Subpunctul 1*

Am generat 1000 de realizări independente a fiecărei repartiții cerute, calculând media si variația,  folosind funcțiile mean, respectiv var.

```r

#Problema 1
#Subpunctul 1

set.seed(124)
binom <- rbinom(1000, 20, 0.2)
binom[1:1000]
mean(binom)
var(binom)

pois <- rpois(1000, lambda = 3)
pois[1:1000]
mean(pois)
var(pois)

expon <- rexp(1000, rate = 1/3)
expon[1:1000]
mean(expon)
var(expon)

norm <- rnorm(1000, mean = 0, sd = sqrt(2))
norm[1:1000]
mean(norm)
var(norm)

```

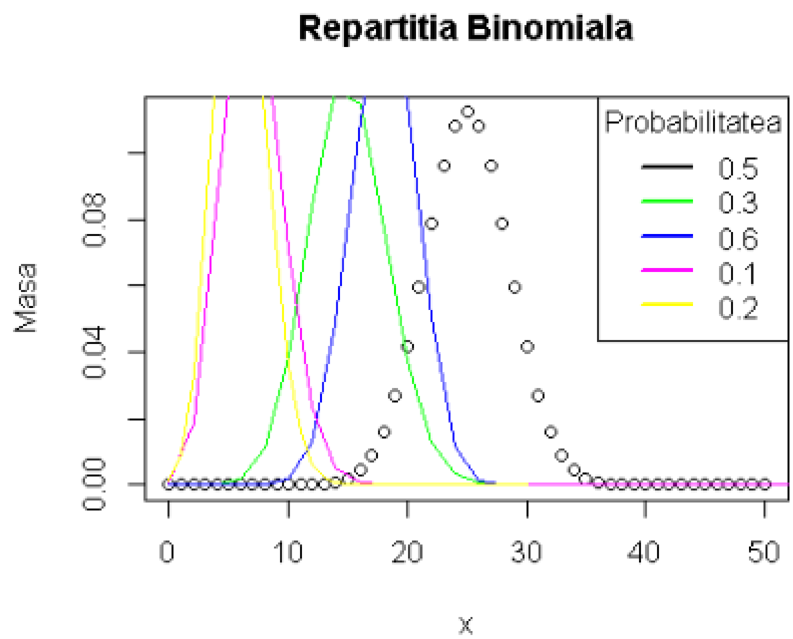
## *Subpunctul* 2

Am folosit funcția plot pentru a desena graficul funcțiilor cerute. Prin intermediul funcției lines am suprapus graficele celor 5 seturi de parametrii. Legenda graficelor am realizat-o folosind funcția legend, cu parametrii corespunzători fiecărei repartiții în parte.

Repartiția Binomială: am folosit funcția dbinom pentru a genera funcția de masă

```r
colors <- c("black", "green", "blue", "magenta", "yellow")
labels <- c("0.5", "0.3", "0.6", "0.1", "0.2")
x <- seq(0,50,by = 1)
plot(x,dbinom(x,50,0.5), main = "Repartitia Binomiala", xlab="x", ylab="Masa")
x <- seq(0,45,by = 2)
lines(x,dbinom(x,50,0.3), col = "green")
x <- seq(0,60,by = 2)
lines(x,dbinom(x,30,0.6), col = "blue")
x <- seq(0,70,by = 2)
lines(x,dbinom(x,70,0.1), col = "magenta")
x <- seq(0,30,by = 1)
lines(x,dbinom(x,30,0.2), col = "yellow")
legend("topright", title="Probabilitatea", labels, lwd=2, lty=c(1, 1, 1, 1, 1), col=colors)
```

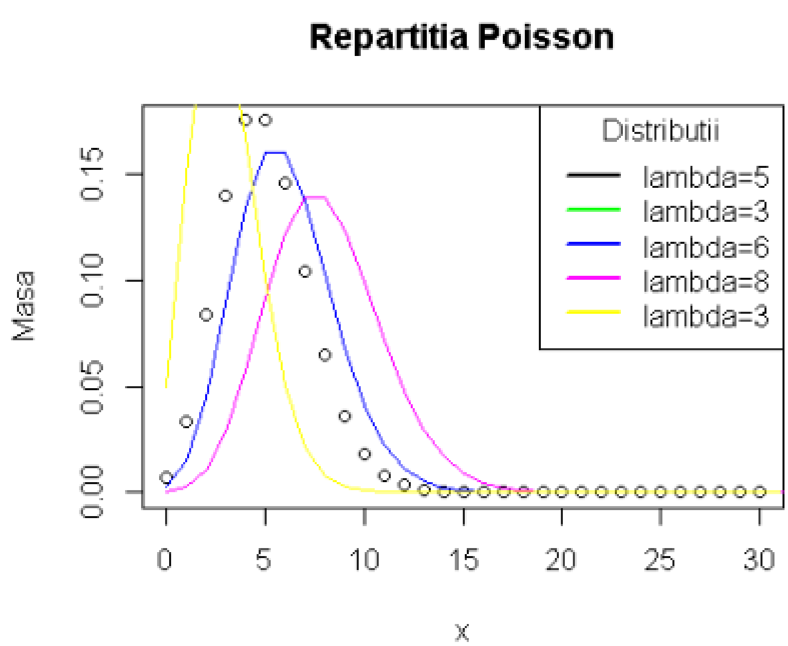
Graficul funcției de masă a celor 5 repartiții binomiale:




Repartiția Poisson: am folosit funcția dpois pentru a determina funcția de masă

```
colors <- c("black", "green", "blue", "magenta", "yellow")
labels <- c("lambda=5", "lambda=3", "lambda=6", "lambda=8", "lambda=3")
plot(0:30, dpois(0:30, 5), main = "Repartitia Poisson", xlab="x", ylab="Densitate")
lines(0:24, dpois(0:24, 3), col = "green")
lines(0:35, dpois(0:35, 6), col = "blue")
lines(0:40, dpois(0:40, 8), col = "magenta")
lines(0:31, dpois(0:31, 3), col = "yellow")
legend("topright", title="Distributii",  labels, lwd=2, lty=c(1, 1, 1, 1, 1), col=colors)
```

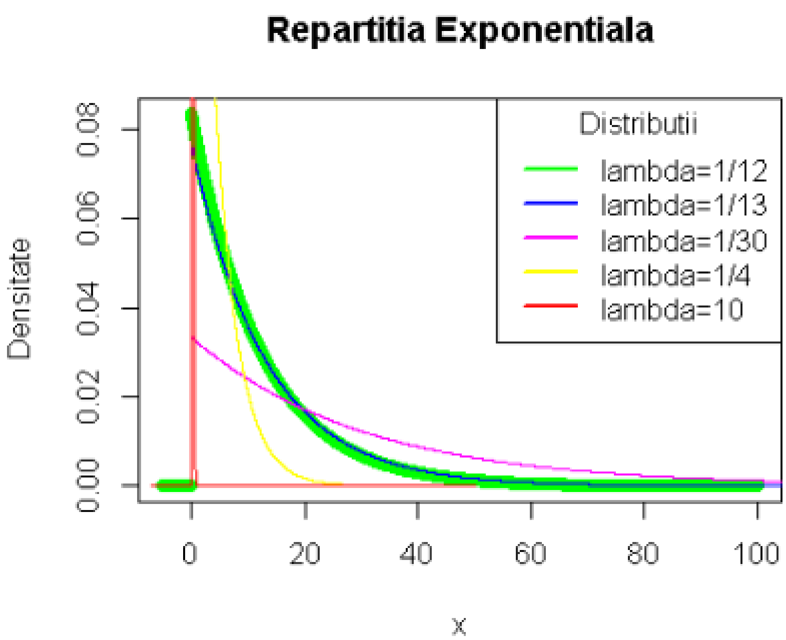
Graficul funcției de masă a celor 5 repartiții Poisson:



Repartiția Exponențială:  am folosit funcția dexp pentru a determina densitatea

```r
colors <- c("green", "blue", "magenta", "yellow", "red")
labels <- c("lambda=1/12", "lambda=1/13", "lambda=1/30", "lambda=1/4", "lambda=10")
lambda <- 1/12
t1 <- seq(-5, 100, 0.01)
plot(t1, dexp(t1,lambda), col = "green", xlab="x", ylab="Densitate", main = "Repartitia Exponentiala")
t2 <- seq(-3, 200, 0.02)
lines(t2, dexp(t2, 1/13), col = "blue")
lines(t2, dexp(t2, 1/30), col = "magenta")
t3 <- seq(-7, 50, 0.01)
lines(t3, dexp(t3, 1/4), col = "yellow")
lines(t3, dexp(t3, 10), col = "red")
legend("topright", title="Distributii", labels, lwd=2, lty=c(1, 1, 1, 1, 1), col=colors)
```

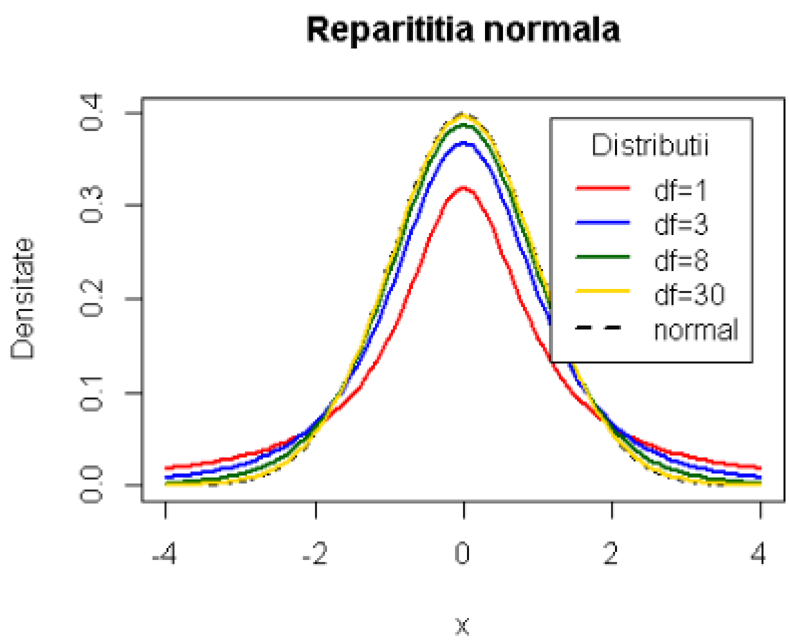
Graficul funcției de densitate a celor 5 repartiții Exponențiale:




Repartiția Normală:  am folosit funcția dnorm pentru a determina densitatea

```r
x <- seq(-4, 4, length=100)
hx <- dnorm(x)

degf <- c(1, 3, 8, 30)
colors <- c("red", "blue", "darkgreen", "gold", "black")
labels <- c("df=1", "df=3", "df=8", "df=30", "normal")

plot(x, hx, type="l", lty=2, xlab="x", ylab="Densitate", main="Reparititia normala")

for (i in 1:4){
        lines(x, dt(x,degf[i]), lwd=2, col=colors[i])
}

legend("topright", inset=.05, title="Distributii", labels, lwd=2, lty=c(1, 1, 1, 1, 2), col=colors)
```

Graficul funcției de densitate a celor 5 repartiții Normale:



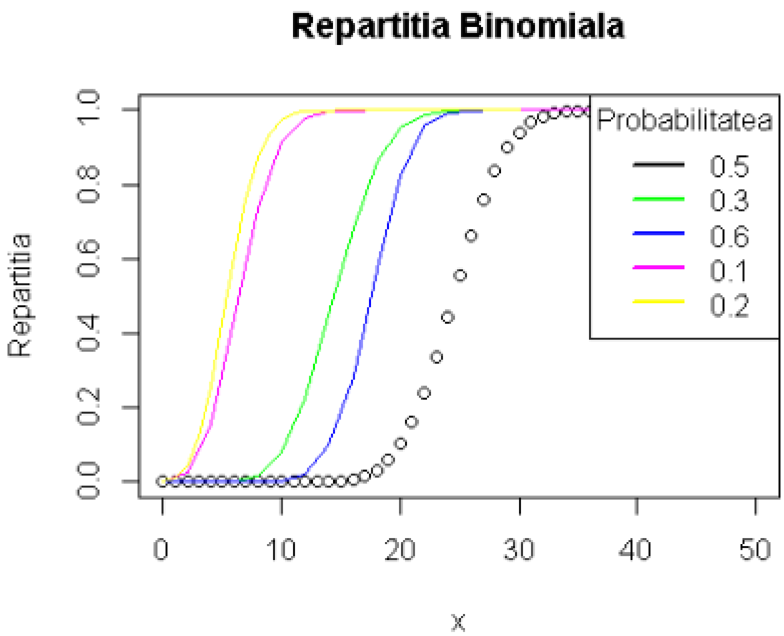

*<u>Subpunctul</u>* 3

Am generat aceleași distribuții cu aceeași parametri, doar ca am folosit alte funcții pentru a determina funcția de repartiție.

Repartiția Binomială: am folosit funcția pbinom pentru a determina funcția de repartiție

```
colors <- c("black", "green", "blue", "magenta", "yellow")
labels <- c("0.5", "0.3", "0.6", "0.1", "0.2")
x <- seq(0,50,by = 1)
plot(x,pbinom(x,50,0.5), main = "Repartitia Binomiala", xlab="x", ylab="Repartitia")
x <- seq(0,45,by = 2)
lines(x,pbinom(x,50,0.3), col = "green")
x <- seq(0,60,by = 2)
lines(x,pbinom(x,30,0.6), col = "blue")
x <- seq(0,70,by = 2)
lines(x,pbinom(x,70,0.1), col = "magenta")
x <- seq(0,30,by = 1)
lines(x,pbinom(x,30,0.2), col = "yellow")
legend("topright", title="Probabilitatea", labels, lwd=2, lty=c(1, 1, 1, 1, 1), col=colors)
```

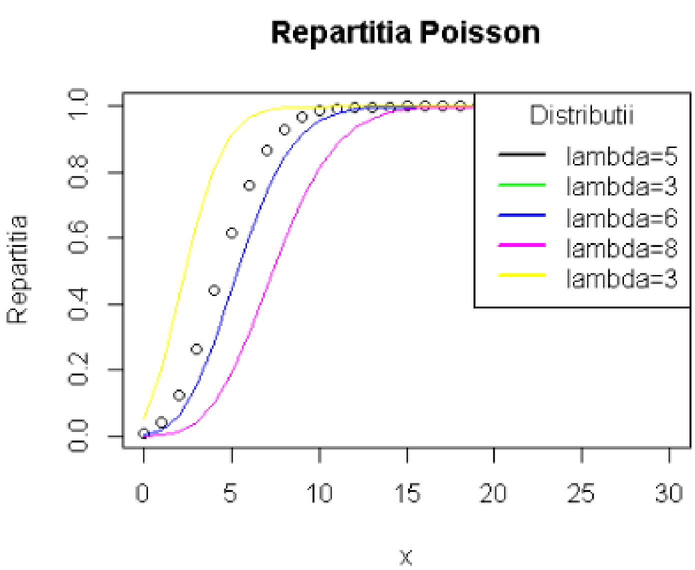
Graficul funcției de repartiție a celor 5 distribuții binomiale:




Repartiția Poisson: am folosit funcția ppois pentru a determina funcția de repartiție

```
colors <- c("black", "green", "blue", "magenta", "yellow")
labels <- c("lambda=5", "lambda=3", "lambda=6", "lambda=8", "lambda=3")
plot(0:30, ppois(0:30, 5), main = "Repartitia Poisson", xlab="x", ylab="Repartitia")
lines(0:24, ppois(0:24, 3), col = "green")
lines(0:35, ppois(0:35, 6), col = "blue")
lines(0:40, ppois(0:40, 8), col = "magenta")
lines(0:31, ppois(0:31, 3), col = "yellow")
legend("topright", title="Distributii",  labels, lwd=2, lty=c(1, 1, 1, 1, 1), col=colors)
```

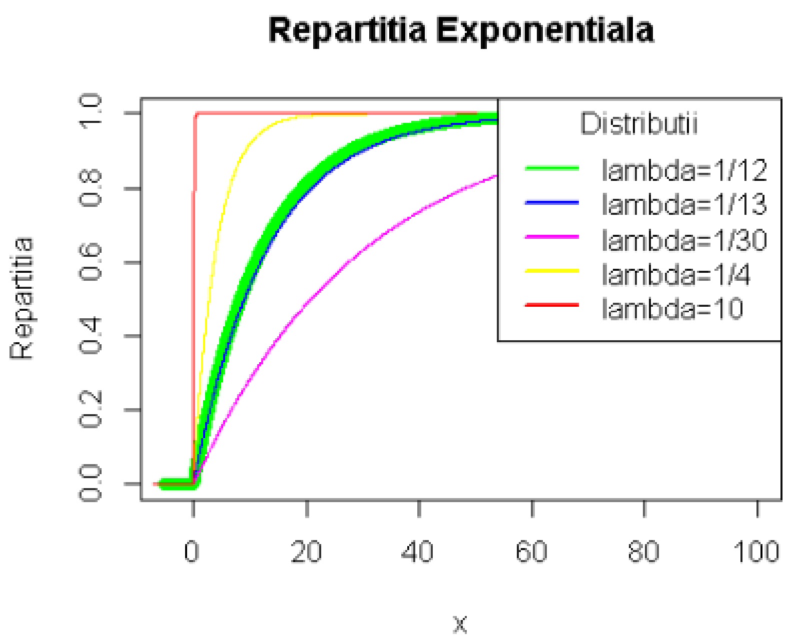
Graficul funcției de repartiție a celor 5 distribuții Poisson:



Repartiția Exponențială: am folosit funcția pexp pentru a determina funcția de repartiție

```
colors <- c("green", "blue", "magenta", "yellow", "red")
labels <- c("lambda=1/12", "lambda=1/13", "lambda=1/30", "lambda=1/4", "lambda=10")
lambda <- 1/12
t1 <- seq(-5, 100, 0.01)
plot(t1, pexp(t1,lambda), col = "green", xlab="x", ylab="Repartitia", main = "Repartitia Exponentiala")
t2 <- seq(-3, 200, 0.02)
lines(t2, pexp(t2, 1/13), col = "blue")
lines(t2, pexp(t2, 1/30), col = "magenta")
t3 <- seq(-7, 50, 0.01)
lines(t3, pexp(t3, 1/4), col = "yellow")
lines(t3, pexp(t3, 10), col = "red")
legend("topright", title="Distributii", labels, lwd=2, lty=c(1, 1, 1, 1, 1), col=colors)
```

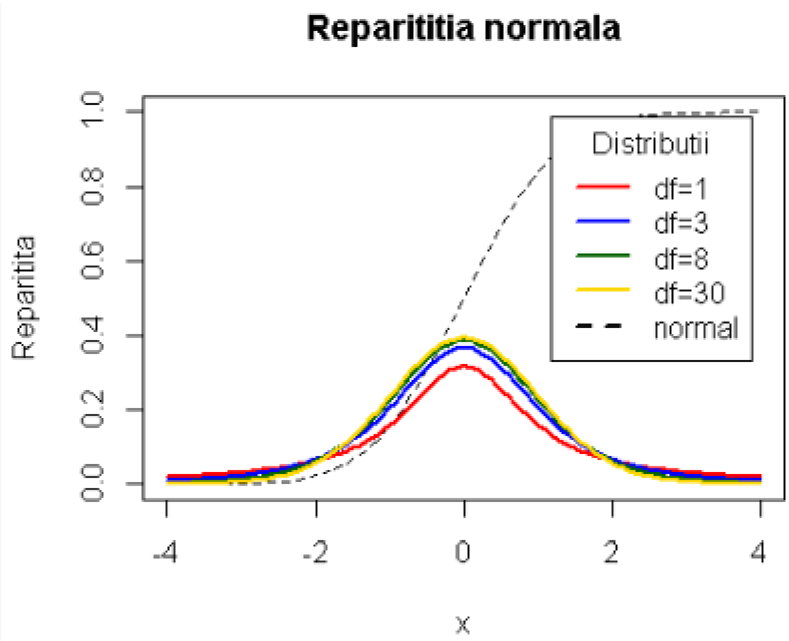
Graficul funcției de repartiție a celor 5 distribuții Exponențiale:




Repartiția Normală: am folosit funcția pnorm pentru a determina funcția de repartiție

```
x <- seq(-4, 4, length=100)
hx <- pnorm(x)

degf <- c(1, 3, 8, 30)
colors <- c("red", "blue", "darkgreen", "gold", "black")
labels <- c("df=1", "df=3", "df=8", "df=30", "normal")

plot(x, hx, type="l", lty=2, xlab="x", ylab="Reparititia", main="Reparititia normala")

for (i in 1:4){
        lines(x, dt(x,degf[i]), lwd=2, col=colors[i])
}

legend("topright", inset=.05, title="Distributii", labels, lwd=2, lty=c(1, 1, 1, 1, 2), col=colors)
```

Graficul funcției de repartiție a celor 5 distribuții Normale:



## Problema 2

*Subpunctul a*

Am definit funcția fgam folosind recurența dată:

```
fgam <- function(a){
  if(a > 1){
    return ((a-1) * fgam(a-1));
  }

  if(a == 1/2)
    return (sqrt(pi));

  if(a == 1)
    return (1);

  f <- function(x){
        return (x^(a-1)*exp(-x));
  }

  temp <- integrate(f, 0, Inf);

  return (temp$value);
}
```

*Subpunctul b*

Am definit funcția fbet folosind recurența dată și funcția fgam definită anterior:

```
fbet <- function(a, b){
  if(a+b == 1 && a > 0 && b > 0)
    return (pi/sin(a*pi));

  return (fgam(a)*fgam(b)/fgam(a+b));
}
```

*Subpunctele c și d*

Am definit funcțiile fprobgammanr, fprobbetanr, fprobnr, cu 1<=nr<=9, în funcție de fiecare cerință, fiecare calculând câte una dintre probabilitățile de la fiecare subpunct cu ajutorul funcțiilor fgam și fbet. După ce am definit aceste funcții, am creat un tabel în care am adăugat rezultatele apelării funcțiilor pentru câte 3 seturi de valori distincte. La final, am calculat aceleași probabilități, de data aceasta folosind funcțiile predefinite în R, pgamma și pbeta, și am trecut rezultatele în același tabel, pe o coloană separată.

Un exemplu: funcția fprobgamma1(a, b), care calculeaza P(X < 3):

```
fprobgamma1 <- function(a, b){
  f <- function(x){
    (x^(a-1)*exp(-x/b))/(b^a*fgam(a));
  }

  integrate(f, 0, 3)$value;
}
```

Tabelul este creat sub forma unei matrice cu 27 de linii și 2 coloane, după cum urmează:

-Prima coloană reprezintă valorile returnate la apelarea funcțiilor create de mine, iar a doua coloană reprezintă valorile returnate de funcțiile predefinite în R (pgamma și pbeta)

-În dreptul fiecărei linii este scrisă cerința pe care o rezolvă, sub forma Pk, cu 1<=k<=9

-La fiecare cerință am folosit câte 3 seturi de date diferite, deci fiecare cerință corespunde pentru 3 linii din tabel

```
table <- matrix(NA, ncol = 2, nrow = 27, byrow = TRUE)
colnames(table) <- c("Functia mea", "Functia din sistem");
rownames(table) <- c("P1", "P1", "P1", "P2", "P2", "P2",
                     "P3", "P3", "P3", "P4", "P4", "P4",
                     "P5", "P5", "P5", "P6", "P6", "P6",
                     "P7", "P7", "P7", "P8", "P8", "P8",
                     "P9", "P9", "P9");
table <- as.table(table);
```

Adăugarea unei linii în tabel se face în felul următor:

```r
table[1, 1] <- fprobgamma1(1, 2);
table[1, 2] <- pgamma(3, 1, 1/2);
```

Afișarea tabelului în consolă:

```
   Functia mea Functia din sistem
P1  0.77686984         0.77686984
P1  0.26424112         0.26424112
P1  0.17335853         0.17335853
P2  0.28579444         0.28579444
P2  0.35202692         0.35202692
P2  0.26516020         0.26516020
P3  0.23865122         0.23865122
P3  0.14105361         0.14105361
P3  0.09989337         0.09989337
P4  0.00000000         0.00000000
P4  0.00000000         0.00000000
P4  0.00000000         0.00000000
P5  0.08554821         0.08554821
P5  0.20905414         0.20905414
P5  0.17793348         0.17793348
P6  0.40572105         0.40572105
P6  0.06593905         0.06593905
P6  0.05075258         0.05075258
P7  0.73318755         0.73318755
P7  0.06824262         0.06824262
P7  0.03013702         0.03013702
P8  0.70186528         0.70186528
P8  0.96870451         0.96870451
P8  0.97815446         0.97815446
P9  0.96881154         0.96881154
P9  0.99893717         0.99893717
P9  0.99883774         0.99883774
```

# Problema 3

## *Subpunctul a*

Am generat prin intermediul funcției frepcomgen matricea repartiției comune incomplete a lui X și Y după următoarea regulă: repartiția marginală a lui X va avea pi = m/(n*m), i=1,2,…,n , iar repartiția marginală a lui Y va avea qj = n/(n*m), j=1,2,…,m. Acest lucru ne permite sa generam matricea astfel încât fiecare element al său să fie egal cu 1/m*n, deoarece suma pe linii și pe coloane din matrice va corespunde cu repartițiile marginale ale lui X, respectiv Y. Apoi am șters elementele de pe diagonală, dându-le valoarea -1, pentru a avea o matrice incompletă. Valorile lui X sunt de la 0 la n-1 și ale lui y sunt de la 0 la m-1.

```r
frepcomgen <- function (n,m)
{
  A = matrix(data = 1/(m*n), nrow = n, ncol = m)

  if(m<n)
  for(i in 1:m) {
    A[i, i] <- -1
  }
  else
    for(i in 1:n) {
      A[i, i] <- -1
    }

  return (A)
}
n <- 5
m <- 6
A <- frepcomgen (n,m)
print(A)
```

```r
x <- 0:(n-1)
y <- 0:(m-1)
print (x)
print (y)

p <- rep(m/(m*n) , n)
q <- rep(n/(m*n) , m)
print(p)
print(q)
```

Matricea va arăta astfel pentru n=5 si m=6:

```
           [,1]        [,2]        [,3]        [,4]        [,5]       [,6]
[1,] -1.00000000  0.03333333  0.03333333  0.03333333  0.03333333 0.03333333
[2,]  0.03333333 -1.00000000  0.03333333  0.03333333  0.03333333 0.03333333
[3,]  0.03333333  0.03333333 -1.00000000  0.03333333  0.03333333 0.03333333
[4,]  0.03333333  0.03333333  0.03333333 -1.00000000  0.03333333 0.03333333
[5,]  0.03333333  0.03333333  0.03333333  0.03333333 -1.00000000 0.03333333
```

## *Subpunctul b*

Am calculate valorile care lipseau, scăzând din repartițiile marginale suma valorilor existente pe linii și coloane.

```
#subpunctul b
fcomplrepcom <- function(A, n, m , x, y, p, q)
{
  if(n > m){
    v <- rowSums(A) + 1
    for(i in 1:m)
      A[i,i] <- p[i] - v[i]

  }
  else{
      v <- colSums(A) + 1
      for(i in 1:n)
        A[i,i] <- q[i] - v[i]
  }

    return(A)
}

B <- fcomplrepcom(A,n,m,x,y,p,q)
print(B)
```

Matricea rezultată este:

```
           [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
[1,] 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333
[2,] 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333
[3,] 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333
[4,] 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333
[5,] 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333
```

## *Subpunctul c*

Aplicând proprietățile Covarianței avem că Cov(3X, 4Y) este 12*Cov(X,Y). Am folosit formula Covarianței : Cov(X,Y) = E[XY] – E[X]E[Y], iar pentru a o aplica a trebuit să calculăm fiecare medie în parte folosindu-ne de repartiții marginale și repartiția comună, aplicând formulele corespunzătoare pentru fiecare:

```
Ex <- 0
for(i in 1:n)
  Ex <- Ex + x[i]*p[i]
print(Ex)

Ey <- 0
for(i in 1:m)
  Ey <- Ey + y[i]*q[i]
print(Ey)

Exy <- 0
for(i in 1:n) {
  for(j in 1:m) {
    Exy <- Exy + B[i,j]*x[i]*y[j]
  }
}
print(Exy)

Cov <- 12 * (Exy -Ex*Ey)
print(Cov)
```

Probabilitatea  P(0 < X < 5 | Y > 4) este egală cu P(0<x<5 , Y>4) /P(Y>4). Pentru a determina P(0<x<5 , Y>4) facem suma valorilor de pe liniile cuprinse între 2 și minimul dintre 5 si n (avem grijă să nu depășim când parcurgem)  și de pe coloane mai mari strict decât 5 (ținem cont că X și Y iau valori de la 0 la n-1, respectiv de la 0 la m-1, atunci cand am generat X și Y), iar pentru a determina P(Y>4)  facem suma valorilor de pe coloane mai mari decât 5.

```
Pxy <- 0
Py <- 0
P <- 0
if(m > 5){
  for(i in 6:m)
    Py <- Py + q[i]
print(Py)

  for(i in  2: min(5,n))
    for(j in 6:m)
      Pxy <- Pxy + B[i,j]
  print(Pxy)

P <- Pxy / Py
}
print(P)
```

Probabilitatea P(X > 3, Y < 7)  este egală cu suma valorilor de pe liniile mai mari decât 5(dacă există), și coloanele mai mici sau egale decât mininimul dintre 7 și m.

```
Pxy <- 0
if(n > 4){

  for(i in 5:n)
    for(j in 1: min(7,m))
      Pxy <- Pxy + B[i,j]
}
print(Pxy)
```

## *Subpunctul d*

Pentru a verifica dacă X și Y sunt independente trebuie să verificăm dacă fiecare valoare a repartiției comune verifică proprietatea:  B[i][j] = p[i]*q[j].

```
fverind <- function(B,n,m,x,y,p,q){
ok <- 1
  for( i in 1:n)
    for(j in 1:m){
      if(B[i,j] != p[i]*q[j])
        ok <- 0
    }
return (ok)
}

ok <- fverind(B,n,m,x,y,p,q)
if( ok == 0)
{
  print("Nu sunt independente")
}
if(ok == 1)
  print("Sunt independente")
```

Două variabile sunt necorelate dacă coeficientul de corelație dat de această formulă $\boxed{\rho(X,Y) = \frac{cov(X,Y)}{D(X)\,D(Y)}}$ este egal cu 0. Coeficientul de corelație este 0 dacă Covarianța este 0, așadar am verificat acest lucru.

```r
fvernecor <- function(Cov)
{
  if(Cov == 0)
    print("Necorelate")
  if(Cov != 0)
    print("Corelate")
}

fvernecor(Cov)
```