Proiect la Probabilitati si Statistica

Budoias Miruna Ruxandra

Pauna Alexandra-Andreea

Grupa 241

Cuprins

PROBLEMA 1

Variabile folosite

```
N = 1000
n = 100 #n repartitie binomiala
p = 0.1 #p repartitie binomiala
lp = 2 #lambda repartitie poisson
le = .3 #lambda repartitie exponentiala
mean = 2 #medie repartitie normala
sd = sqrt(9) #9 = varianta repartitie normala
xd = seq(0, 10)
xc = seq(0, 5, by = 0.5)
```

Functiile mean() si var se folosesc pentru aflarea mediei, respectiv variantei.

Repartitia binomiala

Generarea a N = 1000 variabile aleatoare cu ajutorul repartitiei binomiale:

```
binom = function(n, p) {

  b = c()
  b = rbinom(N, n, p)
  return(b)
}
binom(n, p)
```

```
##    [1] 13 12  9  8 13  4 11 11 11 11  5  7 11  9  9  7  7 13 12  4 13  9 11
##   [24]  7 10  9 13 10 12 11 12  9  8 13  5  8 12 10 10 10 10  6  6  8  9 11
##   [47]  8 12 10 18  6 12  8  7  7  9 11 12  8  5 15  4  9  9 19 11  8 10 11
##   [70] 12 13  7  9 12 11  9 14 10 14 13  8 15 14  5 10  7  9 11 12 12 10  7
##   [93] 10  8 13  6  6 10 12 13  3 14 10 11 11 11 13 12  9 10  7 12  9 14 12
##  [116]  8 15 13 19  5  9  8 12 14 12  8  8 11  7 12 12 13 18 15 10  9  8 16
##  [139] 13 10 11  7  8  8  4  7 11 10 11 18 12 11  8  6 13  5  8  8  8 11  7
##  [162]  6  7 11 12 12 10  9  7  9 14  9  9  9  7  7  8  9  6  9 13 10  4 15
##  [185]  7  8  8  7  6 15  8  8 14 15 11  9  8 11  9  8  8 13  6 13  5 13  9
##  [208] 14  8 12  6  6  8  2  9  5 12  5  6  9  8  6  5  6 13 14  7 11 13 11
##  [231] 15 10  7  9  5 13  8 10 13  4 10 10 11  9 11 11 13  8  4 11  9  7 14
##  [254] 10  8 11 16 13 10  8  8  8  7  6 14 11  8  9 13  6 11  6 13 12 12  8
##  [277] 15  7  6  7  7 15 16  5 14  6 11 10  7  9 11  8  4 12 12  5 10  9  6
##  [300] 10  4  8 13 14 11  8 10 11 10  6 10 17  9  6 11 12 13 13 10  9 11 11
##  [323] 10  8 14 13  7 10 15 12  8 12 12  6  8  3  7  4 13  9 13 12  9  9 15
##  [346] 10 12  6  7 11  5  9  8  9  9 12 14 12  9 10  4  9 11  7  9 12 13 11
##  [369] 11 12  7  9 12 11  9 10 14  6 11  7 16  7 16  8  9  8  3 13 13  5  9
##  [392]  5  9 13  8  6 14 13 15  7 11  8  9  5  7  8  8 13  8  7 18  5 10 10
##  [415] 11  9 11 15  9 10 10 12  7  8  8  8 24 11  6 14 12 12  9  5 16 11 12
##  [438] 13 12  8 17 13 14 14  8 10 12 13  8 11  9  6 13  5 10  6  5 13 10 10
##  [461]  9 10 10 15 11 13  9  5 12 12 10 11  9  9  9  7 13  9  7 16  7 10  6
##  [484] 13 14  7  7  7 12  7  6 15 11  5 10  7 10 11 11 12 16  9  6  9 10  9
##  [507]  9 10  8 16  9 10 10 12 13 11  6 10  9  8 14 10  9  9  5  8  9  9  8
##  [530]  7 14 11  9  7  6 13  8 10 10 11 12 11 13  8 12  9  9 11 18 10 10 12
##  [553] 10 16  9  3 13 11  9  9 10  8 11 12 12 11 10 10  5  8 13  1 10 10  8
```

```
##  [576] 11  8 11 15 18 16  7 11 12  7  8  8 10 13 16  9  9  9  9 10 10  8  7
##  [599] 15  8  7 12 19 10  5  5 11 10 10 19 10  9  5 13  7 10  5  8 10 11 13
##  [622] 11 11 15 11  6  7  9  7 14 12 13 16 13 12  7  8 10 18  7 10  8  7 13
##  [645]  9 12  8  9 12 11 11  9 15 15  7  8  8 11  9  6  9 12 13  5 10  3  5
##  [668]  4  8 11  9 11  5 15  9  6  8 10 13  8 14  9 12 14  9  3 12  8 10  9
##  [691]  9  5  7  7 10  9 10 13  7 11  9 14 14 14  7 11  6 11 12  8 18  7 15
##  [714] 15 20  8  9 10 12 13  5 10  9  8 11  7 11 10 15  5  7 10  6 11 10  7
##  [737]  8 12 11 12 11  6 10 10 11 12 13 10  4 12 10  7 11 13 11  5 10  9 12
##  [760]  7 14  5  5 13  7  9  7  6 13  7 12  8  7  9 13 13  9  9  8 10 14  8
##  [783]  4 17  7 12 16 10  4 14  8  8 11 12  8 12 13 10 12  9  9 14  4  7 11
##  [806] 17 10 11  8  8  8  9 10 11  8  8 14 10  9 12 10  8 13 13 13 10 10 14
##  [829]  8 14 10  9 18 12 13  6 10 15  9  8 11 22 10 10  6  5 10  6  8 10  8
##  [852]  8  8 10  8 16 11 12 14 14  8  8  9 14  8  8  9  9  6 16 14 10  7 10
##  [875]  7  8 12 17 13  9 12  9  6  4  9  6  9 10 10  8  7 11  8 12 12  8 11
##  [898]  8 13 12  7 10 10  8  6 10  7 13  8  5 12 11 10 13 13 11  5  7  8 10
##  [921]  9 12  9  7  9 12 12 11 12  9 10  6 10 12 13 14 13  8 11 11 11  6  6
##  [944] 11  6 13  4  9 11  6 10 11 11 12 12  8  9 10 10  8 14 11 13 10 12 10
##  [967]  8  9 10  9  9  7 10  8 13  7 13 14  9 11  4 13  9  8 15 15 10  7  9
##  [990] 10 13  9 10  9  7 12  4  6 14 11
```

```
cat("[Repartitie binomiala] Media este: ", mean(binom(n, p)), "\n")
```

```
## [Repartitie binomiala] Media este:  9.93
```

```
cat("[Repartitie binomiala] Varianta este: ", var(binom(n, p)), "\n")
```

```
## [Repartitie binomiala] Varianta este:  8.605076
```

Repartitia Poisson

Generarea a N = 1000 variabile aleatoare cu ajutorul repartitiei Poisson:

```
pois = function(l) {

  p = c()
  p = rpois(N, l)
  return(p)
}
pois(lp)
```

```
##    [1] 4 1 2 1 0 4 2 0 1 0 5 0 2 4 2 0 2 5 6 0 0 2 4 3 0 0 3 3 5 1 1 4 0 0
##   [35] 3 0 1 4 0 1 0 0 1 2 0 4 1 3 2 0 4 1 1 1 5 2 1 2 2 3 1 1 4 5 0 1 3 2
##   [69] 1 1 1 2 0 2 2 4 2 3 3 2 1 3 0 0 1 4 0 4 2 1 4 1 1 3 1 2 4 1 1 3 1 2
##  [103] 1 1 2 3 3 3 3 4 3 2 3 1 0 3 2 4 1 1 0 1 4 2 1 1 2 3 1 2 2 2 1 2 1 3
##  [137] 0 1 1 1 1 1 2 3 2 5 1 3 4 3 1 2 4 2 2 3 1 3 1 2 2 5 1 1 2 3 2 2 0 4
##  [171] 0 4 4 1 1 2 3 0 0 2 3 1 2 3 0 3 1 2 2 2 1 2 3 2 1 3 2 2 1 2 1 2 1 2
##  [205] 1 1 4 3 1 4 2 2 0 3 1 3 4 2 4 5 0 3 2 3 1 2 2 0 2 1 0 1 3 0 1 3 1 3
##  [239] 0 1 4 2 5 2 3 3 1 1 7 1 1 1 3 3 1 1 0 2 3 3 3 0 3 2 1 1 4 1 2 0 0 1
##  [273] 0 1 1 4 1 0 4 1 0 1 8 4 3 2 3 0 4 2 1 2 4 4 1 3 3 0 4 2 5 0 1 2 3 2
##  [307] 2 1 2 3 2 4 4 1 2 3 1 3 3 4 2 1 3 1 1 5 4 1 1 3 3 2 0 0 1 2 0 4 1 3
##  [341] 1 3 2 1 0 2 4 3 2 1 1 0 0 5 1 0 3 2 3 0 3 2 4 2 1 2 1 2 2 2 1 2 2 2
##  [375] 1 1 3 1 1 1 0 3 0 5 0 3 5 1 2 9 1 1 2 0 2 4 0 4 1 0 1 3 3 2 5 1 2 7
##  [409] 2 2 3 3 1 1 3 6 0 3 5 5 2 2 4 0 0 3 5 0 2 2 0 4 3 2 3 1 1 0 4 4 2 3
```

```
##  [443] 2 5 1 0 3 1 3 2 4 5 0 3 2 1 2 3 2 1 2 3 4 1 2 1 1 1 3 1 2 4 1 3 2 3
##  [477] 1 5 1 4 0 1 0 2 5 0 1 3 2 1 1 2 2 2 1 1 3 1 1 1 2 2 1 2 3 3 3 4 1 1
##  [511] 2 2 2 1 3 3 0 5 3 4 2 2 1 1 2 0 1 0 0 1 2 1 3 1 3 3 3 0 4 4 3 1 2 0
##  [545] 0 5 4 3 1 0 3 2 1 4 2 3 2 3 2 1 1 2 5 2 1 2 0 4 0 3 1 0 4 2 0 1 3 2
##  [579] 1 1 1 2 5 2 2 0 3 2 2 1 3 2 2 1 2 3 3 4 3 1 0 0 3 0 1 1 2 5 4 0 3 0
##  [613] 2 1 2 2 3 3 2 1 1 2 3 1 1 0 3 4 0 3 1 1 0 3 1 1 2 3 0 0 2 1 4 4 1 2
##  [647] 3 2 1 1 4 2 2 2 3 1 4 2 1 2 2 1 0 1 0 2 2 3 2 2 2 1 0 4 2 2 3 2 1 2
##  [681] 3 3 1 3 1 0 3 2 0 0 1 1 1 3 1 2 1 2 1 5 5 1 2 2 1 3 2 2 3 4 1 0 1 3
##  [715] 1 0 0 3 1 1 5 3 1 2 2 1 4 3 2 0 0 1 2 2 0 3 2 0 3 3 1 0 1 2 1 1 1 5
##  [749] 4 0 1 1 2 2 4 1 2 0 5 0 3 3 3 3 0 0 1 4 2 2 3 4 1 3 1 3 3 2 2 3 4 0
##  [783] 2 1 4 1 1 3 1 2 0 4 3 4 1 2 2 1 0 3 2 2 0 1 1 3 1 1 3 4 3 1 0 2 2 2
##  [817] 1 1 2 0 1 3 1 2 2 2 0 3 1 3 2 3 0 2 1 2 0 3 1 3 3 2 5 3 3 7 3 1 3 2
##  [851] 2 0 2 2 1 3 1 0 0 3 2 3 2 0 1 2 1 1 0 0 2 4 1 2 2 2 0 5 2 4 0 2 2 1
##  [885] 5 1 0 1 1 2 3 2 3 1 0 1 1 3 0 1 2 2 3 1 3 2 2 0 2 0 2 2 6 1 0 1 1 3
##  [919] 3 1 3 0 6 2 1 3 2 1 2 4 3 3 1 1 0 0 4 0 3 4 4 2 1 0 5 1 2 1 2 0 2 3
##  [953] 2 2 2 0 3 4 2 1 5 2 1 2 1 0 2 2 1 1 3 4 2 0 1 1 0 2 2 2 3 1 0 1 2 3
##  [987] 4 1 1 1 4 0 2 1 6 2 0 1 1 1
```

```
cat("[Repartitie Poisson] Media este: ", mean(pois(lp)), "\n")
```

```
## [Repartitie Poisson] Media este:  1.915
```

```
cat("[Repartitie Poisson] Varianta este: ", var(pois(lp)), "\n")
```

```
## [Repartitie Poisson] Varianta este:  2.07071
```

Repartitia exponentiala

Generarea a N = 1000 variabile aleatoare cu ajutorul repartitiei exponentiale:

```
exp = function(l) {

  e = c()
  e = rexp(N, l)
  return(e)
}
exp(le)
```

```
##    [1] 9.660303e-01 1.746029e+01 3.800894e+00 5.365652e+00 1.127096e+00
##    [6] 5.057977e-01 8.746224e-01 1.328652e+01 2.093809e+00 1.259510e+01
##   [11] 2.395577e+00 1.221931e+00 4.487019e+00 8.010773e+00 3.467608e+00
##   [16] 5.396251e+00 8.735990e+00 3.554827e+00 5.342406e+00 8.024832e+00
##   [21] 1.420007e+00 1.950695e+00 4.205940e+00 1.206512e-01 5.209328e-01
##   [26] 2.728045e+00 9.731946e+00 1.120842e+00 3.261637e+00 6.488869e+00
##   [31] 4.881213e+00 4.982378e+00 7.105366e+00 4.257556e+00 9.787978e-01
##   [36] 4.036137e-01 3.402950e+00 7.388468e-01 1.027786e+00 2.272163e+00
##   [41] 2.522231e+00 3.937246e+00 3.719879e+00 1.523405e+00 4.983424e-01
##   [46] 5.896474e+00 4.628010e+00 2.147987e+00 3.899766e-01 4.296912e-01
##   [51] 3.965271e+00 4.979442e+00 8.219180e-01 5.141716e-01 2.172980e+00
##   [56] 1.701479e+00 3.027306e-02 6.647124e-01 1.035234e+01 1.175702e+00
##   [61] 5.176764e-02 3.621526e+00 1.474607e-01 3.173036e-01 1.355647e+01
##   [66] 5.708633e+00 3.040951e+00 4.123618e+00 2.074986e+00 1.993145e-01
##   [71] 2.500315e-01 5.431050e+00 1.032219e+00 5.287877e+00 1.314307e+00
```

```
##   [76] 6.932668e+00 2.888006e-01 1.325276e+00 1.674995e+00 1.201895e+00
##   [81] 1.324081e+00 5.691163e-01 2.910435e+00 2.591996e+00 1.546721e+00
##   [86] 4.033745e-01 3.581048e-01 2.108114e+00 8.559350e+00 5.138077e+00
##   [91] 6.661491e+00 5.650547e-01 3.142737e+00 2.560872e+00 1.646064e+00
##   [96] 3.535919e-01 3.329757e+00 1.682506e+00 6.869104e+00 3.498088e-01
##  [101] 5.099689e+00 4.572197e+00 1.225803e+01 1.875703e+00 3.273673e+00
##  [106] 1.063563e+01 4.191177e+00 7.507660e-01 1.549343e-01 8.329574e+00
##  [111] 3.054462e+00 1.000618e+00 6.081560e-01 3.284527e+00 6.044891e-01
##  [116] 4.233800e+00 7.394689e-01 1.150627e+00 1.231021e+00 3.208838e+00
##  [121] 9.563156e+00 3.235495e+00 2.724245e+00 7.114719e-01 5.400376e+00
##  [126] 6.125250e+00 2.431897e+00 6.293040e-01 1.737334e-01 1.038028e+01
##  [131] 6.834865e+00 6.256753e+00 2.938409e-01 4.990792e+00 8.024595e+00
##  [136] 9.600903e-01 1.087562e+00 1.226026e+00 1.570514e-01 6.719280e+00
##  [141] 9.431604e+00 6.267107e+00 2.229434e+00 2.998938e+00 2.019419e+00
##  [146] 8.731476e+00 1.768919e-01 9.188442e+00 7.600186e+00 9.792467e-01
##  [151] 4.762630e+00 3.698346e+00 3.000306e+00 8.394700e-01 1.608128e+00
##  [156] 9.183280e-01 8.979764e+00 9.334583e-02 1.198379e+00 6.167570e+00
##  [161] 2.659106e+00 6.203430e+00 2.597094e-01 8.715300e+00 1.497837e+01
##  [166] 1.928871e+00 2.665233e+00 2.554213e-01 1.094535e+01 2.120304e+00
##  [171] 1.092883e+01 2.533213e+00 8.403858e-01 7.972472e-01 1.056072e+00
##  [176] 1.486138e+00 5.544278e-01 4.295320e+00 3.073674e+00 1.182854e+00
##  [181] 3.859546e-01 5.623039e-01 2.551769e+00 2.286963e+00 1.814965e+00
##  [186] 4.295245e+00 1.890816e+00 1.269699e+00 4.756869e+00 1.450138e+00
##  [191] 4.476127e+00 4.498241e-01 1.305856e+00 2.304616e+00 3.086435e+00
##  [196] 8.368481e-01 3.794023e+00 4.836874e-01 4.369017e+00 3.917318e+00
##  [201] 1.022859e+00 1.165046e+00 8.578153e+00 3.510353e+00 4.298162e-01
##  [206] 2.021838e+01 5.953929e-01 5.875995e+00 2.270114e+00 1.159124e-01
##  [211] 4.834435e-01 3.014155e+00 2.161424e+00 4.408055e+00 2.812294e-01
##  [216] 4.071064e+00 4.128408e+00 6.259540e-01 2.065955e+00 5.445871e-01
##  [221] 1.597325e+00 4.930486e+00 3.274832e+00 2.730862e+00 7.934398e-01
##  [226] 6.759823e-01 6.689765e+00 7.066533e+00 3.249634e-01 2.609180e+00
##  [231] 2.380859e+00 2.931618e+00 7.161604e-01 5.682205e+00 1.193583e+00
##  [236] 2.381996e+00 3.035466e+00 1.464868e+00 4.922346e+00 2.413197e+00
##  [241] 2.100452e+00 1.989335e+00 1.564326e+00 3.146318e-01 1.189198e+01
##  [246] 6.649202e+00 1.974776e+00 9.603428e-01 1.921610e+00 5.318710e+00
##  [251] 7.290680e+00 7.007590e+00 5.116777e+00 2.787985e+00 9.962106e-01
##  [256] 1.257892e+01 6.690690e-01 7.559895e+00 3.602362e+00 3.708345e+00
##  [261] 1.840876e+00 1.161045e+00 8.143727e+00 1.574495e-01 7.154399e-01
##  [266] 2.211166e+00 2.600913e+00 2.881612e+00 2.423960e+00 5.991948e+00
##  [271] 7.476815e-01 6.032015e-01 4.065232e-01 3.861379e+00 2.016656e+00
##  [276] 2.048398e-01 3.173891e-01 1.283115e+00 4.856831e-01 1.456633e+01
##  [281] 7.672095e-02 1.892753e+00 1.693759e+00 8.053557e-01 2.478525e+00
##  [286] 1.589005e+00 3.450344e-03 1.104513e+01 6.510979e+00 1.566534e+00
##  [291] 1.205646e+00 1.781136e+00 3.937163e+00 9.583120e-01 1.485703e-01
##  [296] 2.632628e+00 3.534919e+00 1.568157e+00 8.344746e-01 1.013682e+00
##  [301] 9.474193e-01 4.274046e-01 1.243140e+01 1.459138e+01 3.565356e-01
##  [306] 1.513063e-01 1.549405e+00 1.713912e+00 4.145015e-01 3.919630e-01
##  [311] 1.602478e+00 6.051172e-01 3.463896e+00 5.608689e+00 2.623825e-01
##  [316] 4.346010e-01 1.599255e+00 2.712647e+00 4.335899e+00 2.912413e+00
##  [321] 2.318943e+00 1.787231e+00 1.508400e+00 6.249918e+00 1.135783e+00
##  [326] 7.994452e+00 1.113924e+00 6.915805e+00 3.091870e+00 2.887776e+00
##  [331] 9.669306e+00 6.817961e+00 3.449688e+00 1.868133e-01 2.895122e+00
```

```
##  [336] 2.556605e+00 4.256098e+00 1.421904e+00 5.881798e-01 2.744788e+00
##  [341] 3.618540e+00 9.070296e-01 1.030374e+00 5.532174e+00 1.069381e-01
##  [346] 1.458609e+01 2.402015e+00 3.232314e+00 1.661947e+00 2.455218e+00
##  [351] 3.630500e+00 1.930935e+00 7.572757e-01 4.159816e-01 2.802640e+00
##  [356] 4.975078e+00 2.034266e+00 4.723938e+00 5.174315e+00 1.137478e+01
##  [361] 5.157114e+00 5.364490e+00 1.383327e+00 4.627672e+00 4.226412e-01
##  [366] 4.553609e-04 4.571724e+00 1.421222e+00 2.645548e+00 7.249013e+00
##  [371] 1.954367e+00 1.725234e+00 1.141770e+00 1.161729e+00 9.396664e+00
##  [376] 3.266789e+00 1.395225e+00 1.552524e+00 9.354640e+00 1.617971e+00
##  [381] 5.223300e-01 6.782330e+00 1.889127e+00 3.485878e+00 5.060337e+00
##  [386] 1.636785e+00 1.490905e+00 3.794500e+00 4.229200e-01 5.224861e+00
##  [391] 6.244749e+00 3.773279e+00 5.478140e+00 3.362466e+00 4.812068e+00
##  [396] 5.682161e+00 1.998827e+00 9.323981e-02 3.675305e-02 3.619521e-01
##  [401] 2.466098e-01 9.061184e-01 7.391211e+00 6.979710e-01 1.724653e-01
##  [406] 1.984348e+00 6.589411e+00 6.131510e+00 1.613207e+01 1.305206e+01
##  [411] 3.978395e+00 1.076325e+01 6.079041e-02 2.316537e+00 6.509043e+00
##  [416] 3.888255e+00 1.224992e+01 2.688997e+00 4.018318e+00 1.436536e+00
##  [421] 7.635028e+00 2.500523e+00 4.410702e+00 2.429937e+00 1.418099e+00
##  [426] 1.269437e+00 3.061159e+00 5.215976e-01 9.349657e+00 1.418810e+00
##  [431] 2.809495e+00 2.327076e+00 1.451622e+00 2.892307e+00 8.559671e+00
##  [436] 6.638986e-01 1.702792e+00 1.777896e+00 5.797377e+00 5.332941e-01
##  [441] 2.779908e+00 7.740118e+00 6.088890e-02 1.966200e+00 9.305140e+00
##  [446] 2.194545e-01 1.572652e+01 2.184104e+00 2.248047e-01 3.323251e+00
##  [451] 1.533837e+01 1.765716e+00 7.337239e+00 1.491089e+00 1.336738e+00
##  [456] 2.673855e+00 3.037539e-01 1.023323e+00 3.685201e+00 5.838434e+00
##  [461] 2.229427e+00 4.741326e+00 3.310703e+00 1.476067e+00 1.365744e+00
##  [466] 1.870280e+00 1.059774e+01 2.968239e-01 2.587433e+00 9.245976e+00
##  [471] 1.833290e+00 1.182027e+00 2.560240e+00 2.115948e+00 2.078493e+00
##  [476] 1.532142e+01 6.752007e+00 2.251409e+00 6.484084e+00 8.634595e-01
##  [481] 1.055897e+01 1.244484e+00 8.140923e-02 3.327690e+00 1.321349e-03
##  [486] 3.479348e+00 1.790826e+00 8.357722e+00 4.603901e+00 2.474862e-01
##  [491] 4.531090e-01 1.721563e+00 8.813797e-01 4.504761e+00 1.498189e+00
##  [496] 3.611550e+00 5.946176e-01 3.257615e+00 5.614116e+00 9.331288e-01
##  [501] 7.282002e+00 2.257011e+00 1.949893e+00 4.370181e-03 8.001741e-01
##  [506] 4.251765e-01 9.021941e-01 6.310337e-01 1.923221e+00 3.493922e+00
##  [511] 4.244539e+00 2.317367e+00 1.166734e+00 2.524021e+00 9.000771e+00
##  [516] 1.060823e-01 2.153459e+00 2.710168e+00 7.868838e-01 6.518168e+00
##  [521] 9.560728e-01 2.277264e+00 6.513610e-01 6.091682e+00 2.049739e+00
##  [526] 1.612794e+00 4.536883e+00 5.533839e+00 6.557997e-04 1.782375e+00
##  [531] 2.295804e+00 2.077568e+00 7.039189e-01 1.787362e+00 7.243561e-01
##  [536] 3.835630e+00 1.043623e+01 1.525775e-01 5.920242e+00 8.311663e-01
##  [541] 9.371899e-01 1.684211e+00 3.431269e+00 9.112936e-01 1.400034e+01
##  [546] 3.618119e+00 2.603549e+00 6.037858e+00 7.671030e-01 2.003998e+00
##  [551] 2.218257e+00 1.353583e+00 3.516159e+00 4.286331e+00 2.222824e+00
##  [556] 2.005567e+00 3.940093e+00 2.793943e+01 1.209054e+01 4.250021e-01
##  [561] 7.285987e+00 7.112242e+00 6.978508e-01 9.529442e-01 1.421922e-01
##  [566] 1.390371e+00 2.125562e+01 2.434584e+00 5.143194e-01 3.308160e+00
##  [571] 3.900800e+00 7.304954e-01 3.624188e+00 1.222023e-01 3.909384e-01
##  [576] 2.031464e-02 1.077043e+00 6.571816e-01 4.046993e+00 2.287206e+00
##  [581] 1.708761e+00 2.972890e+00 6.434084e-01 9.991817e-01 6.583387e+00
##  [586] 1.026721e+00 3.172057e+00 1.909479e+00 1.464328e-01 2.615223e-01
##  [591] 4.894220e+00 3.885896e+00 5.023923e+00 4.862542e+00 1.455937e+00
```

```
##  [596] 1.446162e+00 2.528981e-01 1.527361e+00 4.788278e+00 9.004414e-01
##  [601] 3.673263e+00 2.312585e+00 3.193291e-01 3.606482e+00 6.808198e-01
##  [606] 7.149987e-01 3.166680e+00 1.593701e+00 7.152905e-01 6.321395e+00
##  [611] 2.839781e+00 3.069540e+00 5.216266e+00 2.127445e+00 4.239423e+00
##  [616] 6.366306e-01 5.241336e+00 9.170752e+00 2.363141e+00 4.685646e-01
##  [621] 3.525184e+00 5.973808e+00 4.967515e+00 4.373132e+00 2.574761e+00
##  [626] 2.610616e-01 6.747050e+00 4.251067e-01 4.397368e+00 2.690857e+00
##  [631] 1.165713e+00 1.940520e+00 5.000537e+00 1.137054e+00 2.356418e-01
##  [636] 6.572567e-01 3.543758e-01 6.716209e-01 4.741752e-01 2.113146e+00
##  [641] 2.048200e+00 2.879738e+00 4.388232e+00 8.700801e+00 3.437762e+00
##  [646] 2.238617e+00 2.732773e+00 3.196578e+00 3.374192e+00 2.957147e+00
##  [651] 5.937302e+00 4.883364e+00 3.211709e+00 9.605858e-02 4.259098e+00
##  [656] 1.733260e+00 3.444324e-01 4.627722e+00 5.176432e+00 5.396751e-01
##  [661] 9.591567e-01 2.307275e+00 5.961041e+00 8.029941e+00 2.038144e+00
##  [666] 1.150532e+01 7.380399e+00 6.633629e+00 2.032972e-01 4.134348e+00
##  [671] 2.574211e+00 1.079683e+00 5.288433e+00 2.888189e+00 2.432918e+00
##  [676] 4.113811e+00 1.689377e+00 7.701907e-01 3.285397e+00 4.732422e+00
##  [681] 9.029168e+00 4.392372e+00 1.391988e-01 9.458405e+00 6.489002e+00
##  [686] 1.423698e+00 2.068559e+00 8.598095e+00 2.506591e+00 5.283282e+00
##  [691] 9.661001e+00 5.837081e+00 2.632688e+00 3.955207e+00 5.921678e+00
##  [696] 2.638333e+00 2.552546e+00 2.699622e+00 5.959607e+00 9.385910e+00
##  [701] 2.423110e+00 1.531811e+00 5.729027e+00 8.498637e-01 9.200224e-01
##  [706] 7.703536e-01 1.081239e+00 6.941567e-01 6.370547e+00 7.798085e-01
##  [711] 2.306122e+00 2.378597e+00 1.160407e+00 1.359950e+00 1.107572e+00
##  [716] 3.266528e+00 2.117906e+00 1.933057e+00 1.461991e+00 1.359619e+00
##  [721] 7.378000e+00 1.931267e+00 3.845569e-01 2.007000e+01 7.412784e+00
##  [726] 1.015219e+01 1.834672e+00 5.646242e-01 1.271816e+00 2.075973e+00
##  [731] 3.425830e+00 5.268088e+00 1.680277e+00 2.134293e+00 9.789692e-02
##  [736] 9.454872e+00 7.783409e-02 2.735280e+00 5.950785e+00 2.871122e+00
##  [741] 4.124925e+00 1.268115e-01 1.586575e+00 1.278259e+00 3.906113e+00
##  [746] 2.653109e+00 3.713266e+00 3.094757e+00 8.488276e+00 5.082534e+00
##  [751] 1.029421e+00 6.159900e-01 1.063410e+01 1.171016e+00 3.857087e+00
##  [756] 1.303137e+00 4.943976e+00 8.629160e+00 5.871288e+00 4.679120e+00
##  [761] 2.628374e+00 1.141454e+00 5.879674e+00 5.812180e-01 4.825973e-01
##  [766] 1.064245e+00 2.552287e+00 6.572967e-01 2.029018e+00 4.730334e+00
##  [771] 3.217053e+00 2.442847e+00 3.345985e+00 4.963505e+00 1.698692e-01
##  [776] 5.814369e-01 5.241077e+00 1.825757e+00 5.643375e+00 2.451183e+00
##  [781] 1.250317e+01 7.767070e+00 1.087317e+00 4.847911e+00 4.566494e+00
##  [786] 4.809560e+00 2.857347e-01 1.349453e+00 1.757421e+00 8.616358e-01
##  [791] 2.145290e+00 9.117517e+00 6.515154e-02 1.831738e+00 7.755369e+00
##  [796] 5.734304e-01 9.363871e+00 5.161959e-01 7.654493e+00 2.065074e+00
##  [801] 1.861293e+00 3.281173e+00 7.141627e-01 2.326546e+00 1.450438e+01
##  [806] 2.661574e-01 3.425411e+00 5.518054e+00 6.783765e+00 1.035763e+00
##  [811] 5.432831e+00 8.842499e-02 7.485722e+00 2.979705e+00 7.290209e+00
##  [816] 4.755092e-01 5.549390e-01 3.883475e-01 6.486762e+00 2.131764e+00
##  [821] 2.732632e+00 3.901344e+00 3.903198e+00 7.125697e+00 2.676668e+00
##  [826] 2.398156e-01 5.136376e+00 9.768035e-01 6.892214e+00 1.997354e+00
##  [831] 5.271415e+00 2.843215e-01 4.026395e+00 2.227608e+00 6.713109e-02
##  [836] 2.025444e-01 4.005324e-01 2.391352e+00 3.254843e+00 8.131655e+00
##  [841] 6.122827e-01 4.776097e-01 6.000383e+00 1.385205e+00 2.359530e+00
##  [846] 7.434415e+00 1.586948e+00 5.246656e+00 1.102264e+00 5.923198e+00
##  [851] 2.653786e+00 1.728473e+00 9.139787e+00 1.104535e+00 5.111752e+00
```

```
##  [856] 4.623395e-01 2.682934e+00 6.252115e-01 1.713714e+01 3.069696e+00
##  [861] 4.248117e+00 1.281669e+00 8.282273e+00 5.707912e+00 5.826510e+00
##  [866] 2.319068e+00 9.393473e-01 1.819284e-01 3.009061e+00 8.737457e-01
##  [871] 5.832633e+00 1.761631e+00 1.845627e+00 2.097609e+00 9.943076e+00
##  [876] 5.692115e-01 6.245717e+00 4.247781e-01 1.475900e-01 1.031526e+00
##  [881] 5.658874e+00 1.303200e+00 1.169657e+01 1.395633e+00 5.666872e+00
##  [886] 2.462561e+00 4.291382e-01 5.778738e-01 2.711686e+00 1.736331e+00
##  [891] 3.106202e+00 2.614753e+00 7.552634e-01 1.407892e+01 3.929302e+00
##  [896] 1.329213e+00 7.413743e+00 5.219503e+00 8.159461e-01 5.246396e+00
##  [901] 1.705719e+00 2.064154e+00 2.186595e-01 8.125354e+00 2.777894e+00
##  [906] 5.748132e+00 9.844658e-01 5.793679e-01 3.728745e+00 3.928065e+00
##  [911] 2.354695e+00 3.149101e+00 3.095081e-01 4.272491e+00 4.615789e-01
##  [916] 2.266763e-01 1.905071e+00 2.332234e+00 1.690778e+00 4.823453e+00
##  [921] 1.311943e+01 2.497235e+00 3.614819e+00 3.898245e-01 1.538486e+00
##  [926] 1.891353e+01 1.006686e-01 1.663735e+00 2.363703e+00 4.907170e-01
##  [931] 4.982722e-01 6.492361e-01 2.296957e-01 2.223949e-01 2.866338e+00
##  [936] 6.803604e+00 4.608733e+00 4.581012e+00 2.708128e-01 3.638680e+00
##  [941] 1.832885e-01 4.121140e+00 2.253815e+00 3.258816e-02 1.112030e+00
##  [946] 1.450142e+00 2.011312e+00 9.312514e-02 7.616901e+00 1.024023e+01
##  [951] 7.561667e+00 1.251447e+00 3.871633e+00 9.037305e+00 1.008411e+00
##  [956] 8.890538e-01 5.413461e-01 2.287746e+00 5.621266e+00 2.189454e+00
##  [961] 2.995731e+00 1.917711e+00 6.649322e+00 1.674057e+00 1.303665e+00
##  [966] 2.059940e+00 1.690913e-01 7.845821e-01 1.699550e+00 7.355366e-01
##  [971] 1.784227e+00 6.904115e+00 1.861619e+00 3.816908e+00 8.305684e-01
##  [976] 4.041816e+00 1.398251e+00 1.989853e+00 3.573178e+00 6.477112e+00
##  [981] 7.150345e+00 2.989735e+00 2.183753e+00 7.687445e-01 1.986896e+00
##  [986] 5.450535e+00 3.218021e+00 7.193370e+00 5.763675e+00 8.374398e+00
##  [991] 3.126522e+00 2.819172e+00 4.233604e+00 5.786421e-01 1.168129e+00
##  [996] 3.155286e-01 5.650221e+00 3.759010e-01 4.962007e-01 1.182788e+00
```

```
cat("[Repartitie exponentiala] Media este: ", mean(exp(le)), "\n")
```

```
## [Repartitie exponentiala] Media este:  3.427255
```

```
cat("[Repartitie exponentiala] Varianta este: ", var(exp(le)), "\n")
```

```
## [Repartitie exponentiala] Varianta este:  10.45688
```

Repartitia normala

Generarea a N = 1000 variabile aleatoare cu ajutorul repartitiei normale:

```
norm = function(mean, sd) {

  n = c()
  n = rnorm(N, mean, sd)
  return(n)
}
norm(mean, sd)
```

```
##    [1]  2.3401053244  6.9916625403  2.7099901165 -2.8729504809
##    [5] -0.6547972715  0.1305596923  0.6362818393  2.8832281150
##    [9]  4.8297839813 12.0853098474  4.7350348729  1.3228800263
```

```
##   [13]  0.2047020479  1.2610425007  3.2430939006  2.4300289655
##   [17]  8.8140639312  3.0389719797  3.5407871718  1.4972826969
##   [21]  2.4791878720  4.1042087719  8.3052268250 -3.6080808404
##   [25]  1.6327790686  2.2076991778  2.3013202983  0.4710431280
##   [29]  2.7838493941  0.9110920755  1.7694713369  4.6078902712
##   [33]  2.5208432046  3.1855593761  6.6955978499  2.6776043259
##   [37] -0.2712180004 -6.6819874935  7.3549385148  5.3092237822
##   [41]  3.4518814563  0.6220693895  3.3427407270 -0.9006419535
##   [45]  3.9874133504  2.6160012366 -4.5555621963 -1.5240177075
##   [49]  3.9142240976 -3.9067430451 -4.8024783418  5.3587143855
##   [53] -4.7109193540  0.3832554303  5.8312768823  1.6213193907
##   [57]  8.3025123540 -1.9800973016 -0.0004894438 11.5544292122
##   [61]  0.7253988473  0.7091517010 -2.0451775198  5.7154454369
##   [65]  5.1427045609  7.5746174396  4.5459455705 -0.5484593000
##   [69]  0.9265081758 -3.8687997139 -2.9466389923 -1.0363802309
##   [73] -0.6480124328  2.8744635989  4.5787099772  2.9354046803
##   [77]  5.1993417803 -0.2993875954 -2.7655788343  4.6868653067
##   [81]  4.4826793077 -0.9944273566 -0.3003995015  4.2106950036
##   [85]  3.6381451069 -0.4179261421 -1.7787773506 -0.6082477457
##   [89]  2.1580188275  1.7800507206  2.2769372703  0.8882654486
##   [93]  0.0838379817  2.9408072408  2.5994788059  1.2472283850
##   [97] -1.2378819709  0.2737233497  2.3696526382  3.1706334592
##  [101]  1.4522167266  6.2039015734  5.6018386686  0.9603124968
##  [105]  2.3636138996  5.7805679273  3.6709477720 -0.3106920415
##  [109]  3.1103626695 -0.3884627983  4.6563830051  4.4373801099
##  [113]  3.8520074744  5.3019499127  2.2969200767  3.4802254256
##  [117]  4.3649813856  5.2630616052  4.9998737976 -2.9621572405
##  [121]  6.0754792698  6.0763996360  0.0427826220 -2.1248104159
##  [125]  0.7252474581  1.0792053859 -0.1981405632  2.1203695153
##  [129]  5.4434812759 -1.5130274980  5.4159605915 -2.0139813409
##  [133]  4.9874288976  2.3232354549  7.4578282306  2.9383285246
##  [137]  1.4328378698  1.6070879633  4.4021376241  1.1555545130
##  [141]  0.2064827781  7.0840289311 -0.7411948327  4.0921420031
##  [145]  5.8097547225  2.0355408871  3.8814775247 -2.0394132322
##  [149]  0.8923647973  3.8914746974  5.0417131550 -3.6763918574
##  [153]  3.8953974998 -4.1539485384  5.4666586235 -1.7851957853
##  [157]  0.2279803956  0.0107463219 -4.8429692446  1.3956337492
##  [161] -0.2377093749  6.4801645279 -1.4652034843  1.7350121717
##  [165]  5.2134104886  2.9450545147  3.7637043958  5.1201801154
##  [169]  2.2189632183  1.0109808856  6.1338273585  1.3194453760
##  [173]  1.8113484883  2.8076080741  2.9000706263 -4.3441116928
##  [177]  3.6318502351  3.1979172284 -0.1686287163  2.6896788869
##  [181]  2.9559245683 -1.7714814091  2.7945031432  1.9486202311
##  [185] -3.8338326275 -0.4452194860  4.9988745819  0.5605061289
##  [189]  9.2861528003 -1.7263665016  0.2067566060 -3.2390439229
##  [193]  3.8266909814  1.4860256309  1.3962697001  1.3617022246
##  [197]  4.4472827650 -8.0389710810 -4.5928616990  3.3428997805
##  [201]  3.9599351066 -0.4551647161  0.9737687544  1.8387392753
##  [205]  1.5689805667  6.5613255080  8.8065061745  3.4467648635
##  [209]  1.7350167074 -1.7052573967  3.8159637388  8.3248118397
##  [213] -2.2956130075  5.0557911749  1.8699799729  4.4840634904
##  [217]  4.5051725460  3.8745443717 -0.4941502590  2.1053620984
```

```
##  [221] -2.3225025793 -0.7663816583 -0.3126144061  3.0165541586
##  [225]  2.9379023793  6.0820887215 -0.3478132993  1.6474216781
##  [229]  1.5159412269 -1.9603255843 -2.1715248484  3.0363995439
##  [233]  4.0257545099  4.2164853223  1.4527801730  4.4171769121
##  [237] -0.5778475057  4.9831752852  3.0083502232 -1.5362684020
##  [241] -1.6443223040  0.3585647237 -2.2253824710  0.1853435784
##  [245]  3.2805486740  3.2694492551  1.7229639686  2.0322749429
##  [249] -6.1415662090  2.5196913267  2.0150025576 -3.7904268366
##  [253]  3.8096884042  1.5423220907  0.8636344051  1.3758481405
##  [257] -1.6514713626  0.3268402945  2.2352312140 -1.4226276567
##  [261]  2.1288223312 -0.1654738467  3.3254134341  3.8954788880
##  [265]  0.1889668704  5.7982584834 -0.2334793142  2.1411799830
##  [269]  6.1112217856  3.2909547613  1.4836238460  4.2147267219
##  [273] -0.8058828230  3.7851306282  7.1062398841  6.2563452015
##  [277]  4.0306994796  6.4495795360  4.3911515938  0.4253857347
##  [281]  1.5895472388 -0.5123343529 -0.1196077449  3.5870651223
##  [285]  2.9329406867  3.1356354639  1.3596965545 -0.8322457633
##  [289] -3.1915489943  4.5273147722  0.1693761897 -4.6551442517
##  [293]  0.1420762271 -0.8235850926  4.4939378905  3.2062697800
##  [297]  3.8887395882  0.5499173048  2.6557409475 -2.4262060406
##  [301]  2.8563230536 -4.9832462640  1.7242359921  4.5972309635
##  [305]  2.7841249365  3.1674358946 -1.3874004331  0.7214614171
##  [309]  6.4294377437  5.0271187856  6.1258405486  3.1020855552
##  [313]  2.1778781741  1.9643438862 -2.2100038700 -0.2863322637
##  [317]  0.0620435176 -4.9177584378  7.5411157246  3.6121327312
##  [321]  0.1260679821 -0.6304998635  5.9339943860  4.3268817623
##  [325]  5.5161635440 -4.0402086462  3.1673925545  2.7092957586
##  [329]  3.8104177135  0.5380889724  2.0813378962  3.6832750596
##  [333] -0.0569266200  2.9499647074  3.1405174747 -1.3585700950
##  [337]  4.2997158717  2.0900094181  2.4527528473  4.2123115317
##  [341]  2.9284164351  1.7059075733 -2.7083610095  1.6468517554
##  [345] -0.3713278221  1.6038010087 -1.5947341395 -3.2794504965
##  [349] -1.3193632086  4.3214457753  3.0723342724  4.5285606161
##  [353]  4.2926195114  2.3429607875  2.2250184199 -0.4558207005
##  [357]  6.7955529382 -0.2767116869  4.7381950786  0.8159266080
##  [361] -1.7666122535  2.1399043200  7.6181773794  0.9560516849
##  [365]  5.9418936779  2.3330370876  1.3753160858  0.9757866330
##  [369] -0.1467408638 -1.1763227693  2.9425252975  0.1653255722
##  [373]  4.2637852443  2.3618837206  1.9486293526  2.7400759449
##  [377]  0.1807159195  2.2823667405  2.2150352609  3.7448788382
##  [381] -0.5512695804  1.9582820337  1.2338049777  3.8013053334
##  [385]  2.5032388903  2.9250994880 -1.1285778590  9.5347386769
##  [389]  2.1912534456 -3.8981962832  1.7365222456  6.0453963707
##  [393] -1.4547586566 -0.4555486039 -2.1719996528  3.1589694628
##  [397]  2.9814141098  1.6137126862 -2.7087999157  4.8040785765
##  [401] -4.7327276208 -3.9074205907 -0.8997694569  5.5885610597
##  [405] -0.9759845808  4.5720826151  5.2202144114  3.1813990816
##  [409]  1.0332952159  6.1600783593  1.7263156446  3.5822330069
##  [413] -3.4258735198  3.2002030452 -1.1312458058 -0.1441467509
##  [417]  7.2404982507  2.6538625676  0.4369704863  5.3718943749
##  [421] -1.1789314450  9.2524948663  5.4574510370 -3.5374584550
##  [425]  3.0881995502  0.2593148895 -0.3041011024 -0.1232571441
```

```
##  [429]  0.6821398256  0.8741622211  3.5015875932  1.9891857423
##  [433] -3.2537885828 -0.4228053206  8.5091169972  3.6218418870
##  [437]  4.8449634134  0.2645251746  4.1693609452  1.5706360664
##  [441]  7.1249647870  7.9218645272 -1.2211417566  2.4482180676
##  [445]  3.4920429887  1.7011080239  2.4636631708  1.3113817730
##  [449]  6.0468173881  1.5557776690  0.5297858763  0.7573778574
##  [453] -3.4478799837 -0.0399444042  4.0934570194  0.7691732164
##  [457]  0.2186130079  5.5230281744  6.9852019634 -4.4344232270
##  [461]  2.5018438497 -1.0454262543  1.3417130262 -0.2967746992
##  [465] -0.3669488565 -1.1810212035  8.9844127363  1.2230743711
##  [469]  2.2342589500 -0.2287934950  4.4385415452  2.2312327457
##  [473]  3.6866374574  2.6719353404 -3.7229069126  2.2664968428
##  [477]  4.0914924067  2.4478071414 -0.7212530494  7.2711249925
##  [481] -2.1204071056  2.8655862826  3.8871503878  2.4664402289
##  [485]  0.8310316573  3.3792357000  1.5663877025  1.6165662701
##  [489]  5.4169853131 -3.1932658735  3.8619365899  3.1157263722
##  [493] -3.5580462971  1.2526622346 -1.4043436073  5.0304686213
##  [497]  1.6579508331  1.4296773279  6.2358360788  2.8020279036
##  [501]  6.4755625687  1.4681409328  3.9950620939 -2.9329944565
##  [505]  3.0333203707  1.9025565588  2.1909408148  3.6677995977
##  [509]  2.8926860877  1.3769883474 -1.5685262889  6.6235752994
##  [513]  2.1062490997  3.0827947570  0.4468316943  5.2223555146
##  [517]  5.1068701935  5.8087388649  4.8083219364  2.0633275649
##  [521]  7.0400378607  5.2312899066  0.6235544379  4.1996146893
##  [525]  5.1097203503  0.1555812221  5.7667244042 -3.5155644097
##  [529]  7.3919161259 -0.8894295288  3.3691466423  1.0461008676
##  [533] -0.8576516377  8.7724955786  0.6739279774 -0.1029288127
##  [537]  0.3454611489  3.6864447709 -2.0719823779  1.6286812944
##  [541]  3.4466482159  1.7813972997 -1.1983776433 -0.7914586959
##  [545]  0.8543159082 -1.8458563519 -0.7540235126  2.9408397843
##  [549]  7.3206883465 -0.9401330682  0.1405081539  4.7621331763
##  [553]  8.1814947570  0.9614113769  0.2604304486  2.3885681810
##  [557]  5.8434160564  1.7021525174  3.0027000124  0.7613441414
##  [561] -2.6640209987  4.6555885097  4.5001244727  2.5684281087
##  [565]  1.1461082847 -7.7915337386  0.0122992391  1.7297371225
##  [569]  2.2601982646 -1.1702949330 -1.2654288863  0.6353111295
##  [573] -3.3840048357  5.5314098233  1.7712183289  1.4291178162
##  [577]  1.5612518204 -4.4109231181 -1.0006998440 -1.0002587680
##  [581]  3.2597818935  5.1095391308  5.5216112592  2.9233255885
##  [585] 10.9425349447 -2.5800031054 -1.3718325394  3.1322138414
##  [589]  1.8011232642  3.8310194943  2.6278109454  4.2835096579
##  [593]  3.8025194527  7.5287704239  3.6501411768 -1.2324662889
##  [597]  1.7150321434  4.5759962338  0.7500700088  4.7539311079
##  [601]  5.1313994667  4.4194458317 -2.8153767560 -2.4111599905
##  [605]  1.7803895735  1.6677006698  3.6482848077  2.2823372775
##  [609]  3.7302412844 -1.3057340804  6.3219955629  4.9536211421
##  [613]  3.1222728923  1.2153851920  4.5986800976 -0.1118528081
##  [617]  5.8441215563  0.2561091911  3.2632805321 -1.4439352929
##  [621]  5.9674856519  1.2085685948  3.7471029246  9.0673547180
##  [625]  2.7661659545  0.5584043469  3.6752224757  1.6868970128
##  [629] -2.8725386379  6.6906697692  4.2259104964  3.4369765526
##  [633]  2.3672502890 -0.5780239590  1.9912653111 -3.3733539508
```

```
##  [637] -2.6683022392  3.7245332628  2.8991516118 -3.4014443070
##  [641] -3.7020026103  2.9517222504 -2.4710966018 -0.4853300114
##  [645]  2.6947574521  2.4625549915  4.2080443519  1.3767232468
##  [649] -0.1890902931  6.0465452566  1.1702615749  4.7087621913
##  [653] -2.9744753050  1.7743988026 -1.0852826323  4.0518797525
##  [657]  6.9547335556 -0.6139594613  8.0445115168 -2.5367890725
##  [661]  1.5853065708  4.3012193562  0.5625776567  7.2585348758
##  [665]  3.9599133635  3.5607866935  3.0255733422 -1.3486935345
##  [669]  4.5494415537 -4.8061902589 -2.5633656931  0.6805351604
##  [673] -0.9811380777 -2.2797406946 -4.4393572358  3.4731580506
##  [677]  8.3515158558 -2.9557656712  4.6646392214  1.7718855397
##  [681]  4.3643544218 -2.6806545470  1.3693625241  2.6574973456
##  [685]  2.6018070010  2.9910802760 -0.8249123283  4.0948298653
##  [689] -0.6881876112  0.2169009655 -0.0640617285  5.6203488285
##  [693]  3.6793400066  3.9079303849  2.4185057135  1.5915405846
##  [697] -1.1156323761  2.6154565584  0.9015552972  5.7305083969
##  [701]  5.2266131328  1.9192596002  3.9328880789  4.2924538474
##  [705] -1.9327286160  1.2183111994  0.4656102162  2.9846260796
##  [709]  4.4609807146  2.3191827872  1.2529204365  1.3557374303
##  [713]  0.3189715365  5.5859129432 -0.2926951500  3.8794805721
##  [717]  0.5771173897 -0.1494281523  1.1091161456  5.1606726813
##  [721]  7.1478547453  0.9747898268  2.4226236833  5.4864505566
##  [725] -2.8125737457 -2.0856299816  1.6886851997 -2.5821403566
##  [729]  3.6039580307  8.6695931942  4.8271567296  2.6753847961
##  [733]  1.9662294877  9.1254009415 10.7057691125  4.5124485445
##  [737]  1.2701453587  5.0954071052  0.9070009728 -5.1879231744
##  [741]  3.6719118676  2.1680100413  0.9003198281  0.2026520057
##  [745]  1.9888728307  1.9549269750  2.6413362927  1.9646648804
##  [749]  1.5568153256 -0.9580742884  0.2516221891  5.0638515426
##  [753]  3.0911850664  7.1035693942 -6.3719606231 -3.2293203625
##  [757]  4.2909984604 -0.3525399826  8.1915646008  0.1081767832
##  [761]  6.8489051999 -4.8400175151  6.3401478643  1.0933606397
##  [765]  5.6944515498  5.1200149074 -2.6650799831  0.7029396459
##  [769]  1.3475205688  0.9914010173  2.9130299947  6.0525598971
##  [773]  6.1579981083  1.2487146639 -2.1844177409  4.5032053807
##  [777]  6.2803876887 -0.4368129894 -0.8760983426  2.6633256423
##  [781] -1.1080888145  6.3829126292  2.3426776043  0.7283029768
##  [785] -0.8617849823  6.0571186576  2.8102125644  0.1709315895
##  [789]  3.8203907623  2.0717654734  0.4511866163 -0.1380083231
##  [793]  6.3638208094 -2.5416257125 -2.7323984849  1.9123897429
##  [797]  2.9143668251  5.2185320834  1.0257997409  3.7252404793
##  [801]  2.7029627156 -3.0260045065  1.3230840656  1.1021153655
##  [805]  7.6462906708  1.0192051385  2.4350204600 -0.6884934554
##  [809]  0.3417611218  3.5608221852  5.1478517715  2.6669919421
##  [813]  3.8787376153  0.0055045455  3.1281152597  4.6916292052
##  [817]  4.1944928031  2.7782114432 -0.6165634281  3.1605443488
##  [821] -3.0797717174  0.3574232885  0.6829820429 -1.0128152667
##  [825]  2.9613384655 -2.4970920170  3.7399986954  3.9359879407
##  [829]  2.9454446195  1.5950623621  4.9699378234  3.3311520177
##  [833]  5.8925242004  6.1575649955  4.4572316556  4.3989333620
##  [837] -0.0058668329  1.4088664288  4.4141475152  6.9954942083
##  [841]  2.3071670014  1.5406138483 -0.9427608381  6.4395311118
```

```
##  [845]  1.2178819337  0.7668552019  3.8353700859  4.0335472980
##  [849] -1.0825845885  1.8820294115  6.5868203358  2.9121958827
##  [853]  4.5080637972  3.1229619723  3.3510162587  2.5392611787
##  [857]  5.1991469458  4.4275505007  1.7215861849  2.2948942930
##  [861]  1.8422806779  3.1903564945  5.5579062533  0.1681698371
##  [865]  2.1875249567 -2.0470538137  7.2196642751  1.5370256424
##  [869]  3.5240925089  3.4908834824 -0.4290112443  6.5331691068
##  [873]  3.0953695974  4.8756085842 -1.7114965622  2.6500975423
##  [877]  3.6134554127  0.0206685938  2.4108380770  0.1683167490
##  [881]  4.9466439299  1.1532716833  1.2199138823  7.1881220229
##  [885]  3.0070100528  3.3019867960 -2.5056375731  0.5204208263
##  [889] -2.0843602611  5.4050706141  7.9854253044  2.9171246664
##  [893]  0.8792986317  2.0406774025  4.8902452876  6.3096218480
##  [897] -1.5321418165  3.3031447145  2.0528856143  0.7322510456
##  [901]  2.0584181362 -2.0531299659  4.6046994794 -3.1475884727
##  [905]  3.2427997936  0.1770502248 -3.9635566051 -2.4115682115
##  [909]  1.9420996382 -2.8967819795  1.6912103197 -0.3864737283
##  [913]  3.8813125918 -0.4142555185 -1.1718350675  2.4247867421
##  [917]  3.6315006164 -0.8496197501  5.0563721266  4.8536459049
##  [921]  0.2696506608  5.1163701539  3.1874218807 -0.8030111030
##  [925] -2.1732293896  5.3389479776  0.5627289778  2.0244891412
##  [929]  0.9100748107  0.0197205646 -0.1400620015 -3.4759711246
##  [933]  7.2722735136  0.3488239592  2.8373678073  3.2307134851
##  [937]  0.0248283820  1.0033569645  7.6801519179  1.5253344062
##  [941] -1.7644693096 -0.5322890239  1.5045268474  3.0057977236
##  [945]  1.1719323734  3.8910055273  0.3482716738 -2.9160481736
##  [949] -2.7321522740  3.6542621643  5.0883508032  3.4315534702
##  [953]  6.1067868149  6.4848447030  1.8197787468 -2.6405433560
##  [957]  2.0509870798 -0.7389398402 -1.7987538642 -0.9843469935
##  [961]  1.7660974380 -3.8612165884  5.9669356594  5.4054633095
##  [965] -0.4002569345  4.1000184177  4.5641221177 -0.1170093934
##  [969]  2.1512098152  5.3140129976  2.9642406014 -0.5347084614
##  [973]  3.4035727016  3.0679676010  4.2646028130 -3.9907918484
##  [977] -1.0190809057  2.4301391807  3.0091736651  0.8195596906
##  [981]  2.3775520008  3.9276372864 -4.7625602187  2.4451133357
##  [985]  1.5430800999  1.4122444807  0.7450354113  2.2124805994
##  [989]  5.6444006753  5.9405727520  0.3162055189  1.4703613898
##  [993] -2.6122492608  8.0376436640  1.1333557133  2.9429616533
##  [997]  5.0801226590  1.9465595823 -0.5552258971  1.4350813869
```

```
cat("[Repartitie normala] Media este: ", mean(norm(mean, sd)), "\n")
```

```
## [Repartitie normala] Media este:  2.093079
```

```
cat("[Repartitie normala] Varianta este: ", var(norm(mean, sd)), "\n")
```

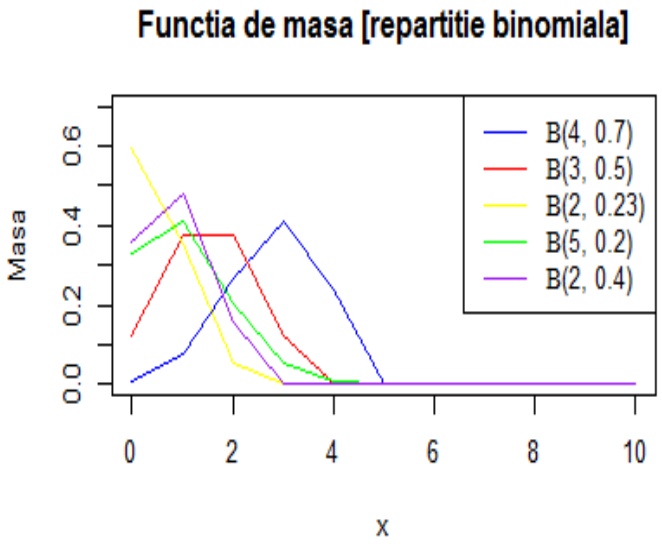
```
## [Repartitie normala] Varianta este:  9.474343
```

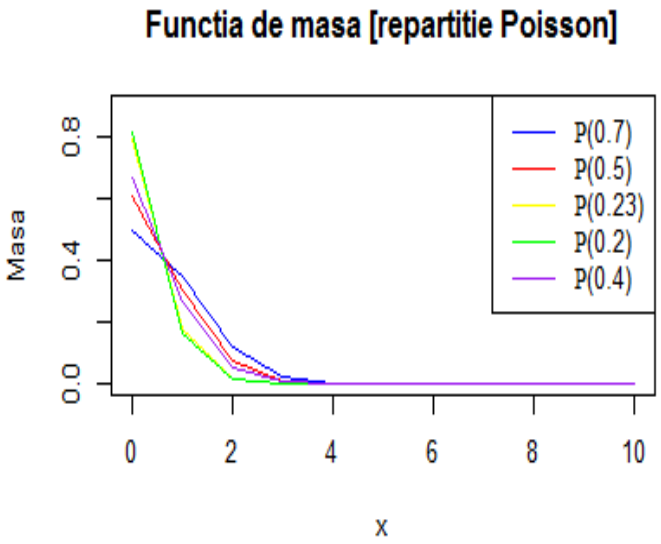
Ilustrarea grafica a functiilor de masa/densitate

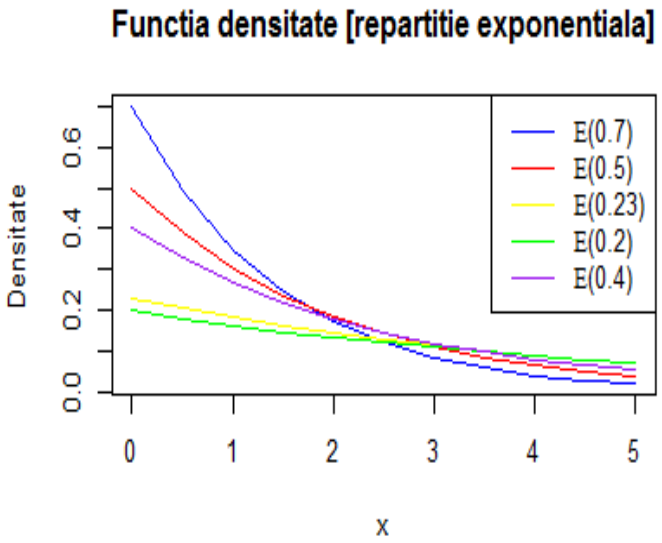
Functia par() foloseste la generarea mai multor grafice in aceeasi imagine. Functia plot() genereaza un grafic, iar functia lines adauga un alt grafic la cel initial. Functia legend() creeaza o legenda. Functiile dbinom, dpois, dexp, dnorm genereaza densitatea/masa fiecarei repartitii.

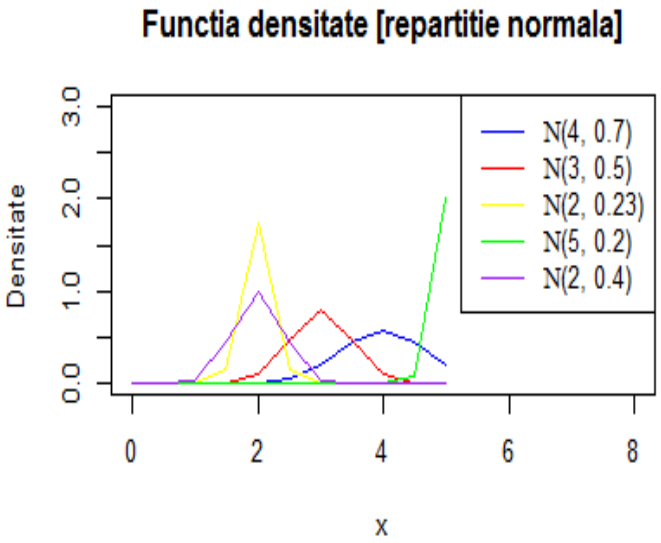
```
exercitiul2 = function() {
  old.par <- par(mfrow=c(2, 2))

  plot (xd, dbinom(xd, 4, 0.7), type = "l", col = "blue", xlab = "x", ylab = "Masa", ylim
= c(0, 0.7), main = "Functia de masa [repartitie binomiala]")
  lines (xd, dbinom(xd, 3, 0.5), col = "red", type = "l")
  lines (xd, dbinom(xd, 2, 0.23), col = "yellow", type = "l")
  lines (xd, dbinom(xd, 5, 0.2), col = "green", type = "l")
  lines (xd, dbinom(xd, 2, 0.4), col = "purple", type = "l")
  legend ("topright", c(expression(paste(Beta, "(4, 0.7)")), expression(paste(Beta, "(3,
0.5)")), expression(paste(Beta, "(2, 0.23)")), expression(paste(Beta, "(5, 0.2)")), expre
ssion(paste(Beta, "(2, 0.4)"))), col = c("blue", "red", "yellow", "green", "purple"), lty
= c(1, 1, 1, 1, 1), lwd = c(1, 1, 1, 1, 1))

  plot (xd, dpois(xd, 0.7), type = "l", col = "blue", xlab = "x", ylab = "Masa", ylim = c
(0, 0.9), main = "Functia de masa [repartitie Poisson]")
  lines (xd, dpois(xd, 0.5), col = "red", type = "l")
  lines (xd, dpois(xd, 0.23), col = "yellow", type = "l")
  lines (xd, dpois(xd, 0.2), col = "green", type = "l")
  lines (xd, dpois(xd, 0.4), col = "purple", type = "l")
  legend("topright", c(expression(paste(Rho, "(0.7)")), expression(paste(Rho, "(0.5)")),
expression(paste(Rho, "(0.23)")), expression(paste(Rho, "(0.2)")), expression(paste(Rho,
"(0.4)"))), col = c("blue", "red", "yellow", "green", "purple"), lty = c(1, 1, 1, 1, 1),
lwd = c(1, 1, 1, 1, 1))

  plot (xc, dexp(xc, 0.7), type = "l", col = "blue", xlab = "x", ylab = "Densitate", main
= "Functia densitate [repartitie exponentiala]")
  lines (xc, dexp(xc, 0.5), col = "red", type = "l")
  lines (xc, dexp(xc, 0.23), col = "yellow", type = "l")
  lines (xc, dexp(xc, 0.2), col = "green", type = "l")
  lines (xc, dexp(xc, 0.4), col = "purple", type = "l")
  legend("topright", c(expression(paste(Epsilon, "(0.7)")), expression(paste(Epsilon, "(0
.5)")), expression(paste(Epsilon, "(0.23)")), expression(paste(Epsilon, "(0.2)")), expres
sion(paste(Epsilon, "(0.4)"))), col = c("blue", "red", "yellow", "green", "purple"), lty
= c(1, 1, 1, 1, 1), lwd = c(1, 1, 1, 1, 1))

  plot (xc, dnorm(xc, 4, 0.7), type = "l", col = "blue", xlab = "x", ylab = "Densitate",
xlim = c(0, 8), ylim = c(0, 3), main = "Functia densitate [repartitie normala]")
  lines (xc, dnorm(xc, 3, 0.5), col = "red", type = "l")
  lines (xc, dnorm(xc, 2, 0.23), col = "yellow", type = "l")
  lines (xc, dnorm(xc, 5, 0.2), col = "green", type = "l")
  lines (xc, dnorm(xc, 2, 0.4), col = "purple", type = "l")
  legend("topright", c(expression(paste(Nu, "(4, 0.7)")), expression(paste(Nu, "(3, 0.5)"
)), expression(paste(Nu, "(2, 0.23)")), expression(paste(Nu, "(5, 0.2)")), expression(pas
```

```
te(Nu, "(2, 0.4)"))), col = c("blue", "red", "yellow", "green", "purple"), lty = c(1, 1,
1, 1, 1), lwd = c(1, 1, 1, 1, 1))
  par(old.par)
}
exercitiul2()
```







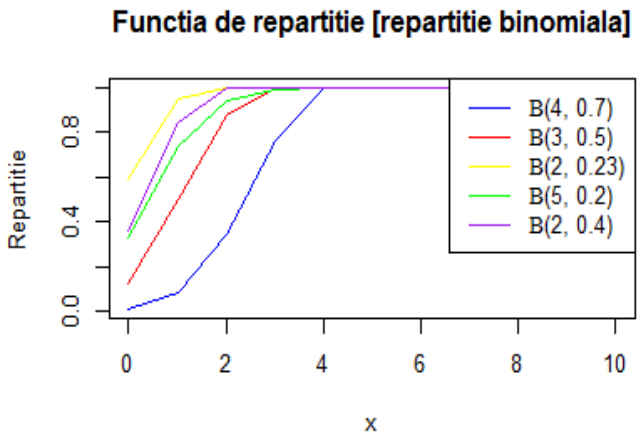


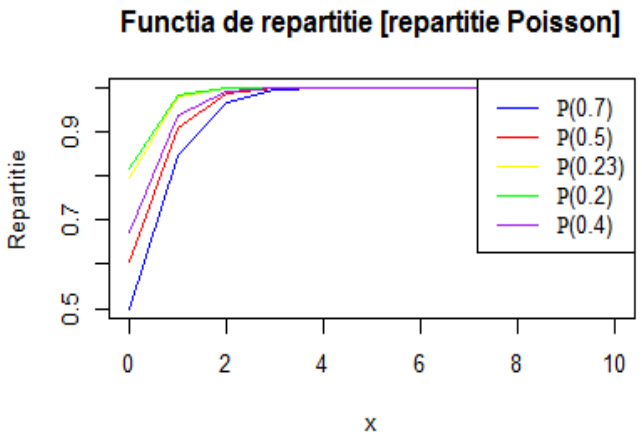
Functiile de repartitie pentru cele 5 seturi de parametrii diferiti

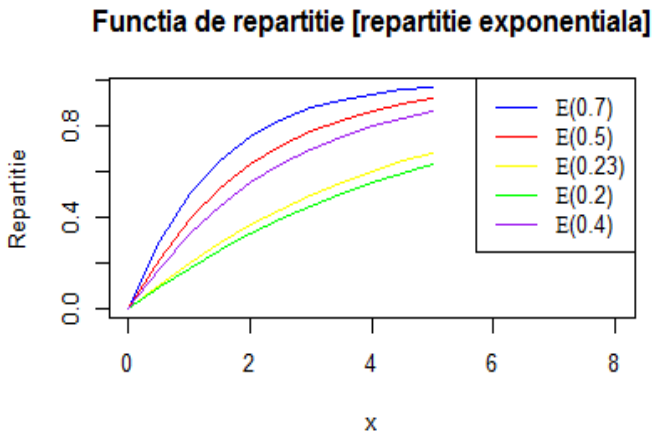
Functiile pbinom, ppois, pexp, pnorm genereaza repartitiile corespunzatoare.

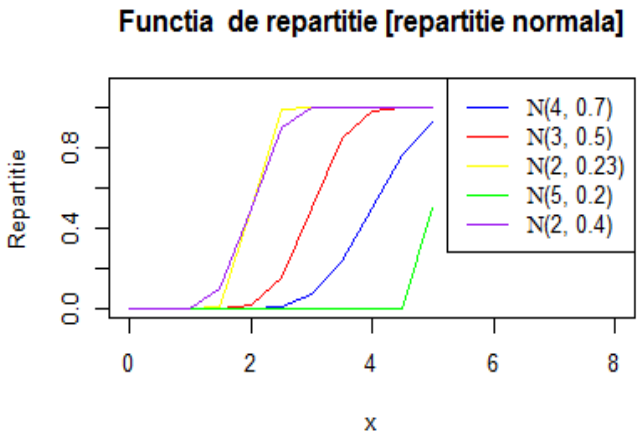
```
exercitiul3 = function() {
  old.par <- par(mfrow=c(2, 2))

  plot (xd, pbinom(xd, 4, 0.7), type = "l", col = "blue", xlab = "x", ylab = "Repartitie"
, main = "Functia de repartitie [repartitie binomiala]")
  lines (xd, pbinom(xd, 3, 0.5), col = "red", type = "l")
  lines (xd, pbinom(xd, 2, 0.23), col = "yellow", type = "l")
  lines (xd, pbinom(xd, 5, 0.2), col = "green", type = "l")
  lines (xd, pbinom(xd, 2, 0.4), col = "purple", type = "l")
  legend ("topright", c(expression(paste(Beta, "(4, 0.7)")), expression(paste(Beta, "(3,
0.5)")), expression(paste(Beta, "(2, 0.23)")), expression(paste(Beta, "(5, 0.2)")), expre
ssion(paste(Beta, "(2, 0.4)"))), col = c("blue", "red", "yellow", "green", "purple"), lty
= c(1, 1, 1, 1, 1), lwd = c(1, 1, 1, 1, 1))

  plot (xd, ppois(xd, 0.7), type = "l", col = "blue", xlab = "x", ylab = "Repartitie", ma
in = "Functia de repartitie [repartitie Poisson]")
  lines (xd, ppois(xd, 0.5), col = "red", type = "l")
  lines (xd, ppois(xd, 0.23), col = "yellow", type = "l")
  lines (xd, ppois(xd, 0.2), col = "green", type = "l")
  lines (xd, ppois(xd, 0.4), col = "purple", type = "l")
  legend("topright", c(expression(paste(Rho, "(0.7)")), expression(paste(Rho, "(0.5)")),
expression(paste(Rho, "(0.23)")), expression(paste(Rho, "(0.2)")), expression(paste(Rho,
"(0.4)"))), col = c("blue", "red", "yellow", "green", "purple"), lty = c(1, 1, 1, 1, 1),
lwd = c(1, 1, 1, 1, 1))

  plot (xc, pexp(xc, 0.7), type = "l", col = "blue", xlab = "x", ylab = "Repartitie", xli
m = c(0, 8), main = "Functia de repartitie [repartitie exponentiala]")
  lines (xc, pexp(xc, 0.5), col = "red", type = "l")
  lines (xc, pexp(xc, 0.23), col = "yellow", type = "l")
  lines (xc, pexp(xc, 0.2), col = "green", type = "l")
  lines (xc, pexp(xc, 0.4), col = "purple", type = "l")
  legend("topright", c(expression(paste(Epsilon, "(0.7)")), expression(paste(Epsilon, "(0
.5)")), expression(paste(Epsilon, "(0.23)")), expression(paste(Epsilon, "(0.2)")), expres
sion(paste(Epsilon, "(0.4)"))), col = c("blue", "red", "yellow", "green", "purple"), lty
= c(1, 1, 1, 1, 1), lwd = c(1, 1, 1, 1, 1))

  plot (xc, pnorm(xc, 4, 0.7), type = "l", col = "blue", xlab = "x", ylab = "Repartitie",
xlim = c(0, 8), ylim = c(0, 1.1), main = "Functia  de repartitie [repartitie normala]")
  lines (xc, pnorm(xc, 3, 0.5), col = "red", type = "l")
  lines (xc, pnorm(xc, 2, 0.23), col = "yellow", type = "l")
  lines (xc, pnorm(xc, 5, 0.2), col = "green", type = "l")
  lines (xc, pnorm(xc, 2, 0.4), col = "purple", type = "l")
  legend("topright", c(expression(paste(Nu, "(4, 0.7)")), expression(paste(Nu, "(3, 0.5)"
)), expression(paste(Nu, "(2, 0.23)")), expression(paste(Nu, "(5, 0.2)")), expression(pas
te(Nu, "(2, 0.4)"))), col = c("blue", "red", "yellow", "green", "purple"), lty = c(1, 1,
1, 1, 1), lwd = c(1, 1, 1, 1, 1))
```

```
  par(old.par)
}
exercitiul3()
```









Aproximarea functiei de repartitie si de masa a binomialei

```
aproximarea = function(n, p){
  probBin = NULL
  densBin = NULL
  probPois = NULL
  probNorm = NULL
  probNoFC = NULL
  probCP = NULL

  q = 1 - p

  for(k in 1:10){
    probBin[k] = pbinom(k, n, p)
    densBin[k] = dbinom(k, n, p)
    ps = rpois(n, n * p)
    probPois[k] = length(ps[ps <= k]) / n
```

```
    nm = rnorm(n, n * p,sqrt(n * p * q))
    N = (k - mean(nm)) / sd(nm)
    probNorm[k] = pnorm(N)

    Nc = ((k + 0.5) - mean(nm)) / sd(nm)
    probNoFC[k] = pnorm(Nc)

    r = abs((k + 1) * (1 - p)) / abs(p * (n - k))
    a = 1 / (9 * (n - k))
    b = 1 / (9 * (k + 1))
    S = a + b * (r ^ (2 / 3))
    m = 1 - a
    c = (1 - b) * (r ^ (1 / 3))
    probCP[k] = pnorm((c - m) / sqrt(S))

  }
  cbind(densBin, probBin, probPois, probNorm, probNoFC, probCP)
}
```

n = 25 p = 0.05

```
##             densBin   probBin   probPois  probNorm   probNoFC    probCP
##  [1,] 3.649863e-01  0.6423759   0.68    0.4733693  0.6868201   0.6452200
##  [2,] 2.305177e-01  0.8728935   0.92    0.6747482  0.7954858   0.8733858
##  [3,] 9.301589e-02  0.9659094   0.96    0.9668103  0.9922728   0.9651268
##  [4,] 2.692565e-02  0.9928351   1.00    0.9999736  0.9999992   0.9922684
##  [5,] 5.951987e-03  0.9987870   1.00    0.9998997  0.9999883   0.9985791
##  [6,] 1.044208e-03  0.9998312   1.00    0.9999997  1.0000000   0.9997790
##  [7,] 1.491726e-04  0.9999804   1.00    0.9999970  0.9999995   0.9999705
##  [8,] 1.766518e-05  0.9999981   1.00    1.0000000  1.0000000   0.9999966
##  [9,] 1.756187e-06  0.9999998   1.00    1.0000000  1.0000000   0.9999997
## [10,] 1.478894e-07  1.0000000   1.00    1.0000000  1.0000000   1.0000000
```

n = 25 p = 0.1

```
##             densBin   probBin   probPois  probNorm   probNoFC    probCP
##  [1,] 1.994161e-01   0.2712059   0.24   0.2031114   0.2898639   0.2706080
##  [2,] 2.658881e-01   0.5370941   0.40   0.3882023   0.5188958   0.5383546
##  [3,] 2.264973e-01   0.7635914   0.68   0.6904117   0.8590071   0.7647348
##  [4,] 1.384150e-01   0.9020064   0.88   0.8727431   0.9555870   0.9021393
##  [5,] 6.459368e-02   0.9666001   0.96   0.9660813   0.9852869   0.9662328
##  [6,] 2.392358e-02   0.9905236   1.00   0.9978155   0.9993024   0.9901932
##  [7,] 7.215049e-03   0.9977387   1.00   0.9995168   0.9998638   0.9975741
##  [8,] 1.803762e-03   0.9995425   1.00   0.9999615   0.9999915   0.9994841
##  [9,] 3.785674e-04   0.9999210   1.00   0.9999998   1.0000000   0.9999050
## [10,] 6.730087e-05   0.9999883   1.00   0.9999999   1.0000000   0.9999848
```

n = 50 p = 0.05

```
##             densBin   probBin   probPois  probNorm  probNoFC    probCP
##  [1,] 0.2024867770  0.2794318   0.18    0.2700333  0.4042998  0.2787915
##  [2,] 0.2611013704  0.5405331   0.54    0.3504656  0.4854873  0.5419320
```

```
##  [3,] 0.2198748382  0.7604080   0.74    0.6597806  0.7805109  0.7617348
##  [4,] 0.1359752289  0.8963832   0.86    0.8514955  0.9084585  0.8966284
##  [5,] 0.0658406372  0.9622238   0.98    0.8991660  0.9420236  0.9618574
##  [6,] 0.0259897252  0.9882136   0.98    0.9858225  0.9940889  0.9878222
##  [7,] 0.0085981046  0.9968117   1.00    0.9947921  0.9976524  0.9965871
##  [8,] 0.0024323585  0.9992440   1.00    0.9999915  0.9999987  0.9991503
##  [9,] 0.0005974214  0.9998414   1.00    0.9999845  0.9999963  0.9998103
## [10,] 0.0001289172  0.9999704   1.00    0.9999988  0.9999998  0.9999617
```

n = 50 p = 0.1

```
##          densBin    probBin   probPois   probNorm   probNoFC     probCP
##  [1,] 0.02863208  0.03378586   0.08     0.01654694  0.03006247  0.03347043
##  [2,] 0.07794290  0.11172876   0.12     0.07722463  0.11451055  0.11092527
##  [3,] 0.13856515  0.25029391   0.34     0.13970507  0.19999518  0.24973031
##  [4,] 0.18090450  0.43119841   0.46     0.27197177  0.35428172  0.43140627
##  [5,] 0.18492460  0.61612301   0.62     0.44468647  0.54337080  0.61685407
##  [6,] 0.15410383  0.77022684   0.76     0.65785811  0.74731569  0.77089036
##  [7,] 0.10762807  0.87785492   0.92     0.82856185  0.87724122  0.87812696
##  [8,] 0.06427788  0.94213279   0.88     0.90746228  0.94216308  0.94206842
##  [9,] 0.03332927  0.97546206   1.00     0.96301052  0.97753255  0.97526407
## [10,] 0.01518333  0.99064540   1.00     0.99360233  0.99681914  0.99046300
```

n = 100 p = 0.05

```
##          densBin    probBin   probPois   probNorm   probNoFC     probCP
##  [1,] 0.03116068  0.03708121    0.03    0.03858378  0.05958831  0.03669114
##  [2,] 0.08118177  0.11826298    0.16    0.13131399  0.17934121  0.11735451
##  [3,] 0.13957568  0.25783866    0.26    0.14695558  0.20546279  0.25722355
##  [4,] 0.17814264  0.43598130    0.46    0.34404938  0.43116745  0.43621282
##  [5,] 0.18001783  0.61599913    0.61    0.61148072  0.70148832  0.61680670
##  [6,] 0.15001486  0.76601398    0.86    0.66444773  0.74768467  0.76677287
##  [7,] 0.10602554  0.87203952    0.88    0.84093823  0.89299352  0.87238911
##  [8,] 0.06487089  0.93691041    0.92    0.89931008  0.93248803  0.93688116
##  [9,] 0.03490130  0.97181171    0.98    0.94304203  0.96364024  0.97160682
## [10,] 0.01671588  0.98852759    0.98    0.98173833  0.98940506  0.98831516
```

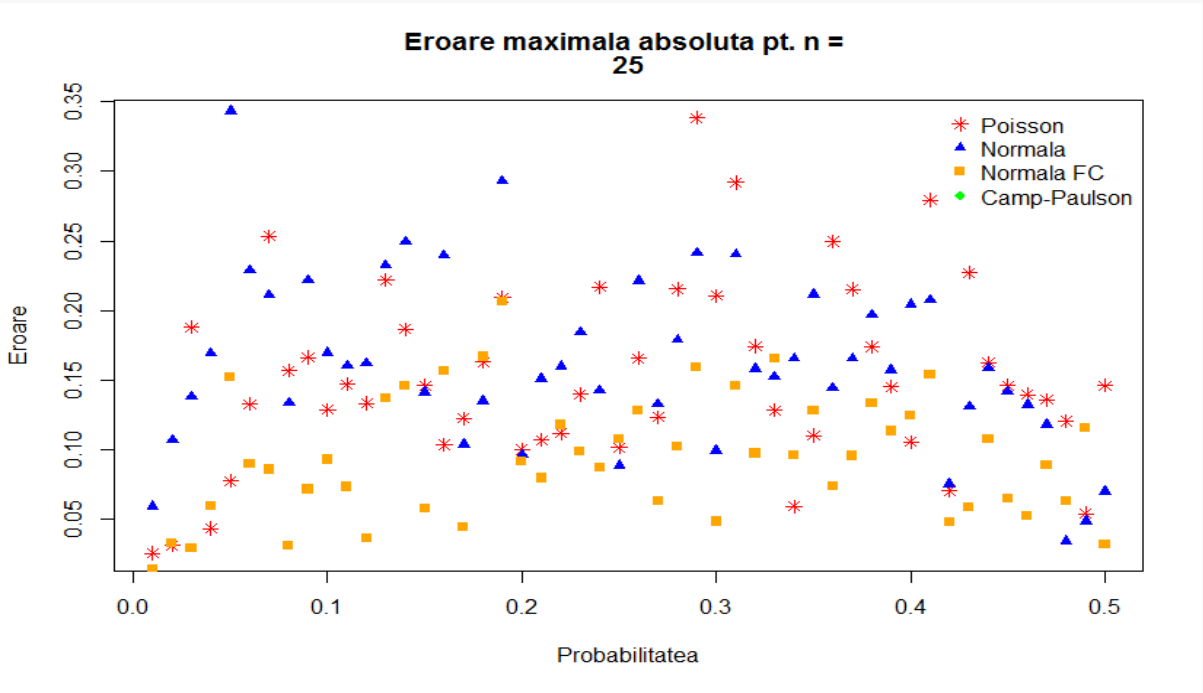
n = 100 p = 0.1

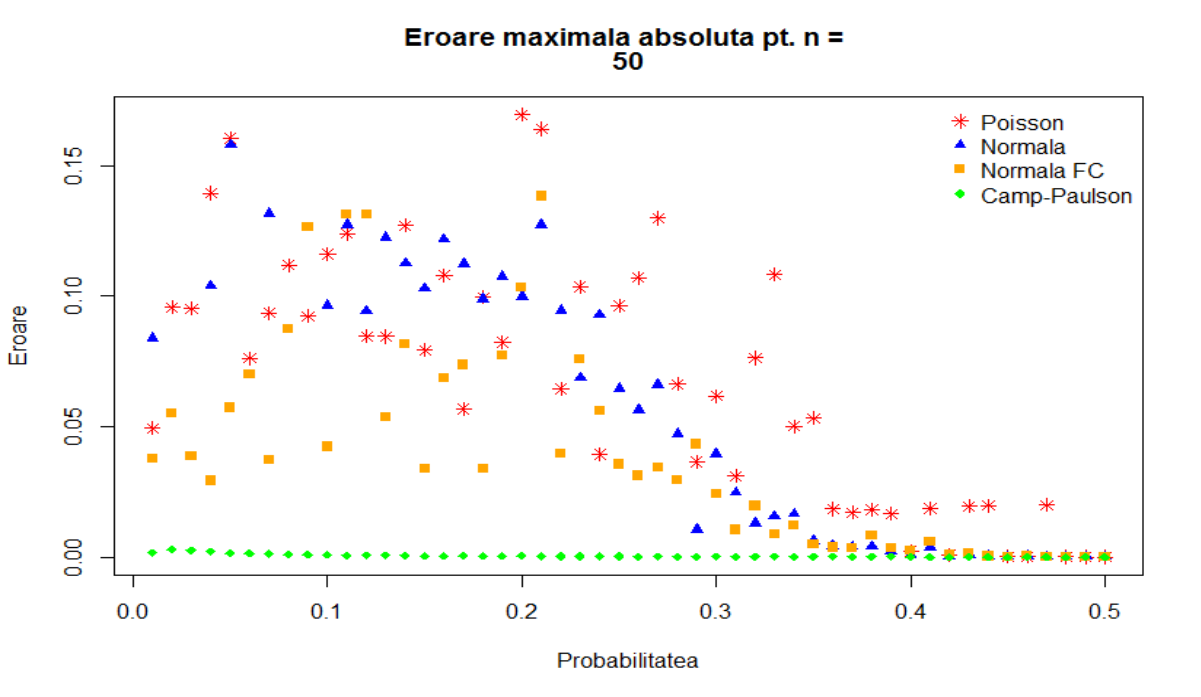
```
##            densBin    probBin    probPois    probNorm    probNoFC        probCP
##  [1,] 0.0002951267  0.0003216881   0.00    0.001843607  0.003078370    0.0003768871
##  [2,] 0.0016231966  0.0019448847   0.00    0.005777595  0.008927185    0.0020472627
##  [3,] 0.0058916025  0.0078364871   0.03    0.019822617  0.028259575    0.0079372432
##  [4,] 0.0158745955  0.0237110827   0.02    0.023439712  0.034195644    0.0237016956
##  [5,] 0.0338658038  0.0575768865   0.13    0.061752663  0.083775090    0.0573669670
##  [6,] 0.0595787289  0.1171556154   0.13    0.076561092  0.102973984    0.1167730189
##  [7,] 0.0888952464  0.2060508618   0.18    0.143245884  0.185346374    0.2056576427
##  [8,] 0.1148230266  0.3208738884   0.32    0.235477588  0.286619151    0.3206667053
##  [9,] 0.1304162771  0.4512901654   0.41    0.386516916  0.457825060    0.4513689123
## [10,] 0.1318653468  0.5831555123   0.54    0.515517793  0.579786038    0.5834688363
```

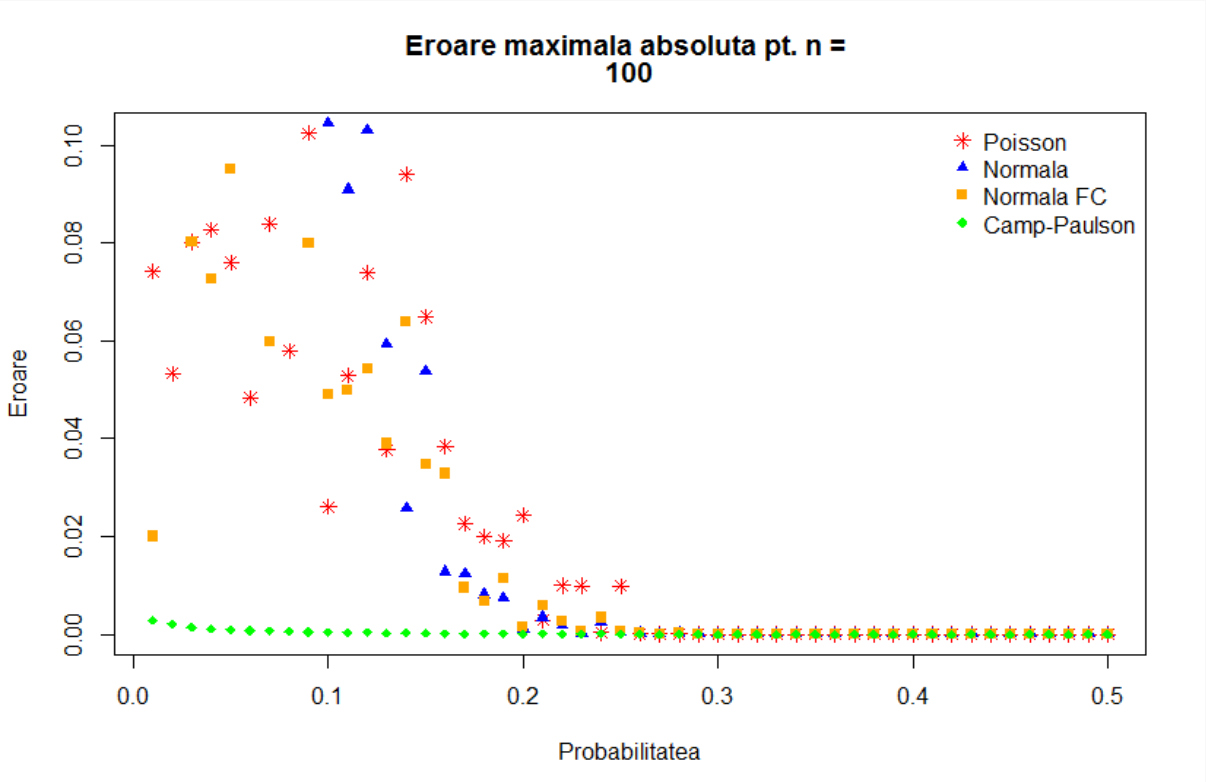
Eroarea maximala absoluta

```
eroarea = function(n){
  p = seq(0.01, 0.5, 0.01)

  probBin = NULL
  probPois = NULL
  probNorm = NULL
  probNoFC = NULL
  probCP = NULL

  errPois = NULL
  errNorm = NULL
  errNoFC = NULL
  errCP = NULL

  for(i in seq_along(p)){
    maxP = 0
    maxN = 0
    maxNc = 0
    maxCP = 0;

    q = 1 - p[i]

    for(k in 1:10){
      probBin = pbinom(k, n, p[i])

      ps = rpois(n, n * p[i])
      probPois = length(ps[ps <= k]) / n

      nm = rnorm(n, n * p[i], sqrt(n * p[i] * q))
      N = (k - mean(nm)) / sd(nm)
      probNorm = pnorm(N)

      Nc = ((k + 0.5) - mean(nm)) / sd(nm)
      probNoFC = pnorm(Nc)

      r = abs((k + 1) * (1 - p[i])) / abs(p[i] * (n - k))
      a = 1 / (9 * (n - k))
      b = 1 / (9 * (k + 1))
      S = a + b * (r ^ (2 / 3))
      m = 1 - a
      c = (1 - b) * (r ^ (1 / 3))
      probCP = pnorm((c - m) / sqrt(S))

      maxP = max(maxP, abs(probPois - probBin))
      maxN = max(maxN, abs(probNorm - probBin))
      maxNc = max(maxNc, abs(probNoFC - probBin))
      maxCP = max(maxCP, abs(probCP - probBin))
    }
    errPois[i] = maxP;
    errNorm[i] = maxN;
```

```
    errNoFC[i] = maxNc;
    errCP[i] = maxCP;
  }
  plot(p, errPois, xlab = "Probabilitatea", ylab = "Eroare", col = "red", pch = 8)
  title(main = c("Eroare maximala absoluta pt. n = ", n))
  points(p, errNorm, col = "blue", pch = 17)
  points(p, errNoFC, col = "orange", pch = 15)
  points(p, errCP, col = "green", pch = 18)
  legend("topright", pch = c(8, 17, 15, 18), c("Poisson", "Normala", "Normala FC", "Camp-
Paulson"), col = c("red", "blue", "orange", "green"), bty="n")
}
```







Observam ca aproximarea normala cu factor de corectie este mai precisa decat aproximarea normala,aproximarea Poisson are cele mai mari erori dintre toate cele 4 studiate, iar aproximarea Camp Paulson estecea mai precisa. In momentul in care numarul de observatii creste, toate functiile tind sa aproximeze din cein ce mai bine binomiala B (n, p).

PROBLEMA 2

Justificarea teoretica a simularii cuplului

Plecam de la doua variabile repartizate uniform pe [-1, 1] demonstram :

$$X_1 \sim U[-1,1] \Rightarrow P(X_1 = x_1) = \frac{1}{2}, \forall x_1 \in [-1,1]$$

$$X_2 \sim U[-1,1] \Rightarrow P(X_2 = x_2) = \frac{1}{2}, \forall x_2 \in [-1,1]$$

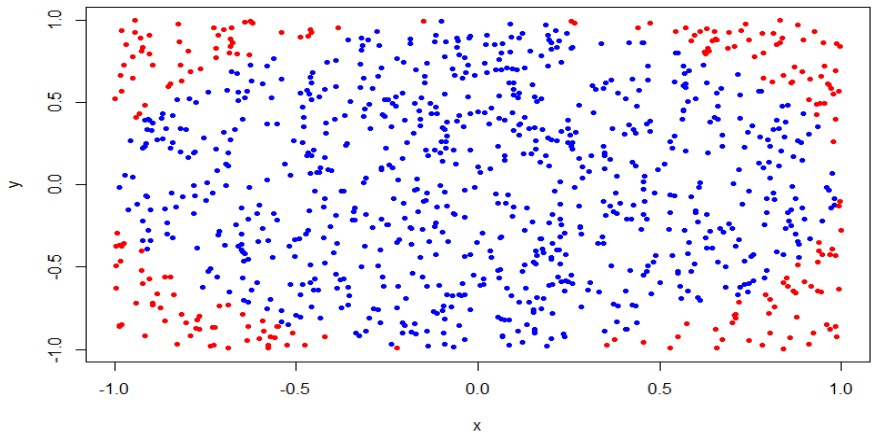
$$P\big((X_1, X_2) = (x_1, x_2)\big) = P(X_1 = x_1, X_2 = x_2) = P(X_1 = x_1)P(X_2 = x_2) = \frac{1}{4}, \forall x_1, x_2 \in [-1,1]$$

$$P(X_1 = x_1, X_2 = x_2) = \frac{1}{4} = \frac{1}{Aria([-1,1]^2)} \Rightarrow (X_1, X_2) \sim U([-1,1]^2)$$

Metoda acceptarii si respingerii

Punctele reprezentate cu albastru apartin discului de unitate D(1), iar cele reprezentate cu rosu nu apartin.

```
acc_reject = function(){
  x <-c(runif(1, min = -1, max = 1))
  y <-c(runif(1, min = -1, max = 1))
  for (i in 2:N)
  {
    x <-c(x, runif(1, min = -1, max = 1))
    y <-c(y, runif(1, min = -1, max = 1))
  }

  plot(x, y, col = ifelse(x * x + y * y <= 1, "blue", "red"), pch = 20)
}
```



Simularea punctelor prin coordonate polare

```
polar_coords = function(){
  theta <-runif(1, min = 0, max = 2 * pi)
  rho <-runif(1, min = 0, max = 1)
  x <-c( rho * cos(theta))
  y <-c(rho * sin(theta))

  for (i in 2:N)
  {
    theta <- c(theta, runif(1, min = 0, max = 2 * pi) )
    rho <- c(rho, runif(1, min = 0, max = 1))
    x <- c(rho * cos(theta))
    y <- c(rho * sin(theta))
  }

  plot(x, y)
  circle = seq(-1, 1, by = 0.01)
  lines(circle, sqrt(1 - circle^2), col ="red")
  lines(circle, -sqrt(1 - circle^2), col ="red")
}
```