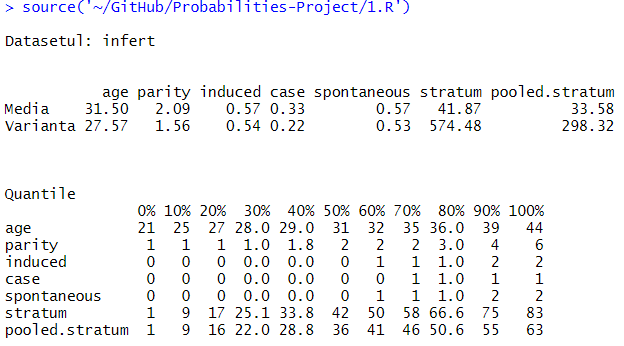
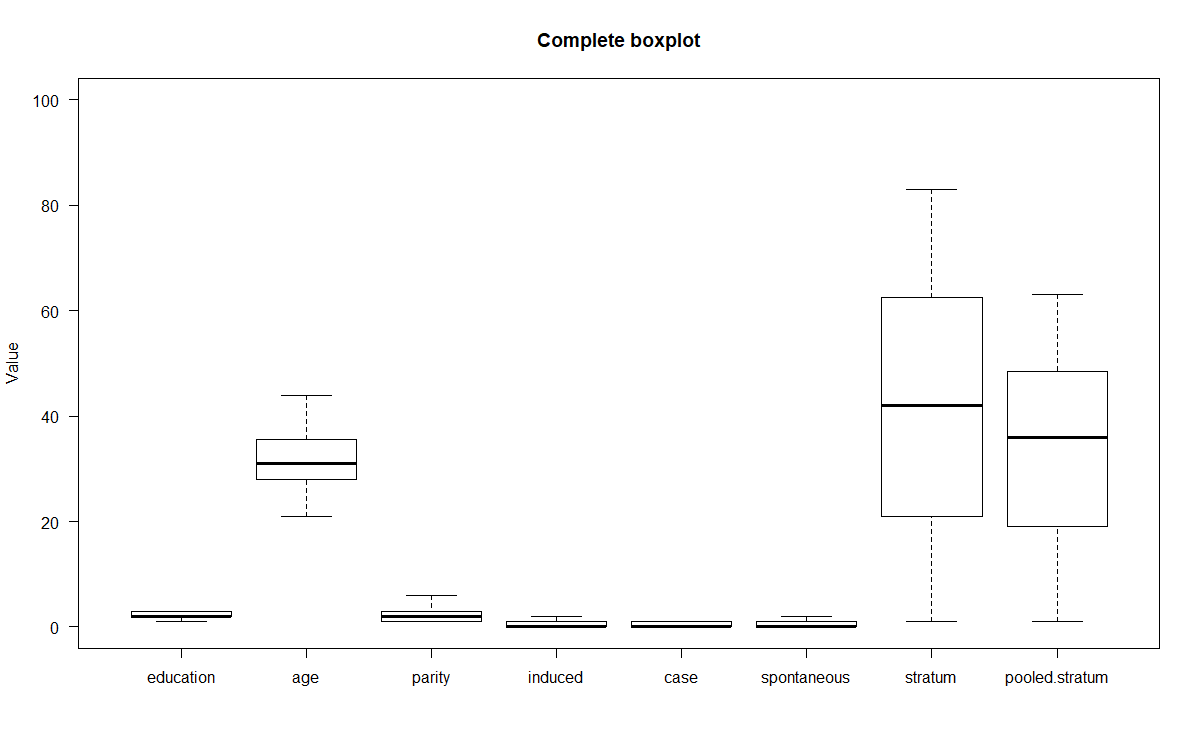
Proiect la Pobabilități și Statistică

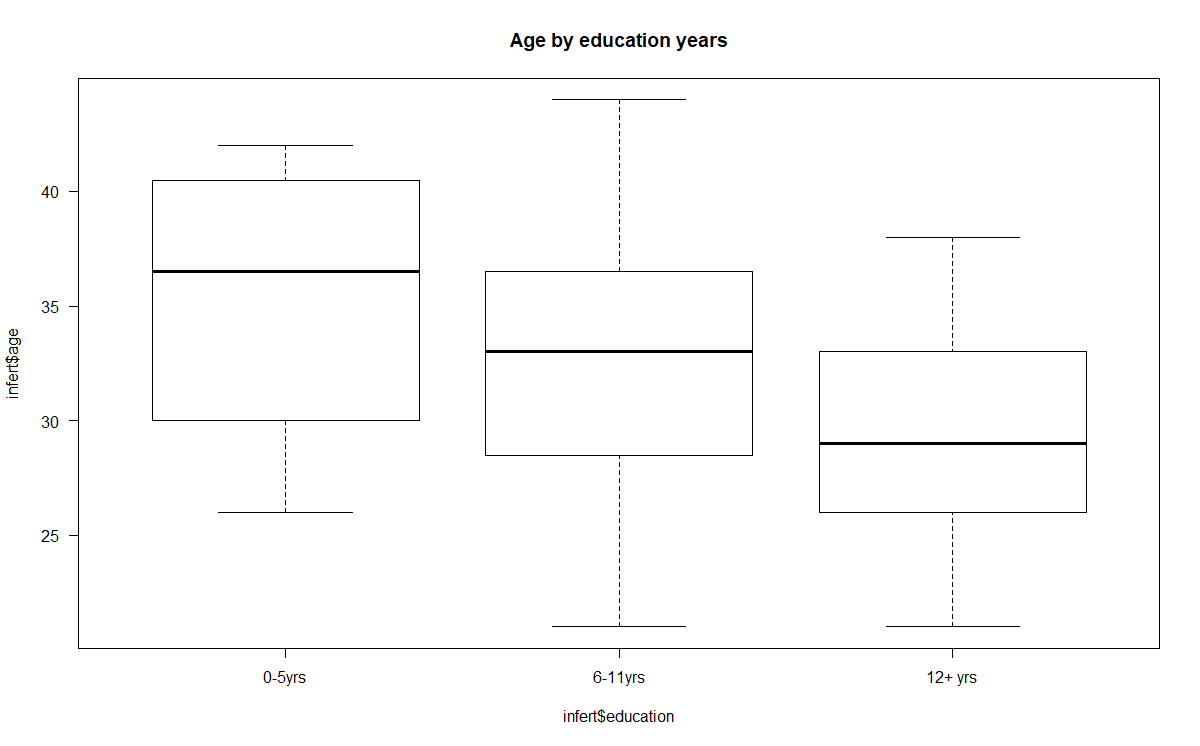
1. Folosind setul de date **Infert** efectuaţi operaţii de statistică descriptivă pentru variabilele din acest set de date (medie, varianţa, quartile, boxplot, interpretări).

Setul de date Infert conține date despre femei care sunt mame (care au dat naștere cel puțin unui copil) și acum sunt infertile.



****

Se poate observa cum varsta, startum si pooled.startrum fluctuează cel mai mult, restul variabilelor luand un numar foarte mic de valor.

****

Se poate observa că o data cu avansarea studiilor, mediana vârstei scade drastic, iar cum Q1 si Q3 sunt mai apropiate de Q2, putem trage concluzia ca deviația scade.

2. Ca regresie simpla am ales density plot-ul, grafic care determina daca variabila este aproape de normala.

Am ales sa determinam desitatea varstei:

density(age)

plot(density(age), main="Age density", xlab="AGE", las=1)

Ca regresie multipla vom afisa un diagnostic intre varsta, paritate, cazuri de avorturi induse si spontane:

m2<-lm(age~parity+induced+case+spontaneous)

plotModel(m2)

Deoarece setul nostru de date analizează infertilitatea, vom adăuga două variabile noi, s\_births și f\_births, reprezentând numărul de sarcini duse până la capăt, respectiv numărul de sarcini eșuate, variabile relevante pentru acest domeniu.

s\_births<-sample(x=c(0,1,2,3,4), size=length(infert[[1]]), replace=TRUE)

f\_births<-sample(x=c(0,1), size=length(infert[[1]]), replace=TRUE)

Vom afisa totodata si corelatia dintre acestea si varsta:

cor(age,s\_births)

cor(age,f\_births)

Dupa care generam o regresie simpla pentru s\_births:

scatter.smooth(x=age, y=s\_births, main="age ~ successful births")

m3<-lm(age~parity+induced+s\_births+f\_births)

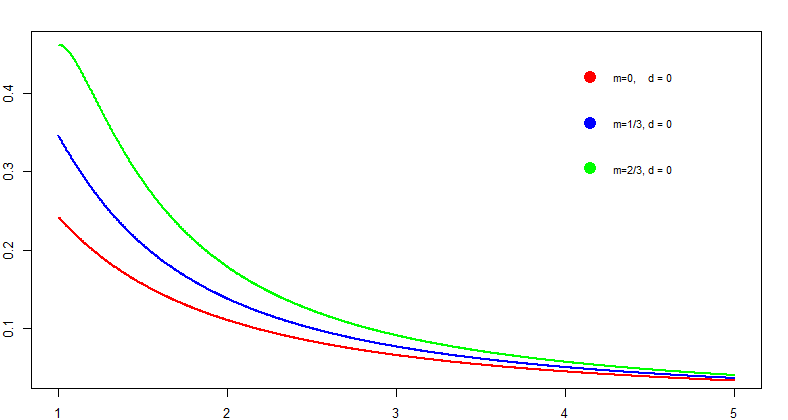
plotModel (m3)

Pentru ca este un eveniment rar ca o variabilă din setul de date să fie în relație cu doar o altă vatiabila, folosim regresia multipla, deoarece aceasta explică o variabilă dependentă folosind mai multe variabile independente simultan:

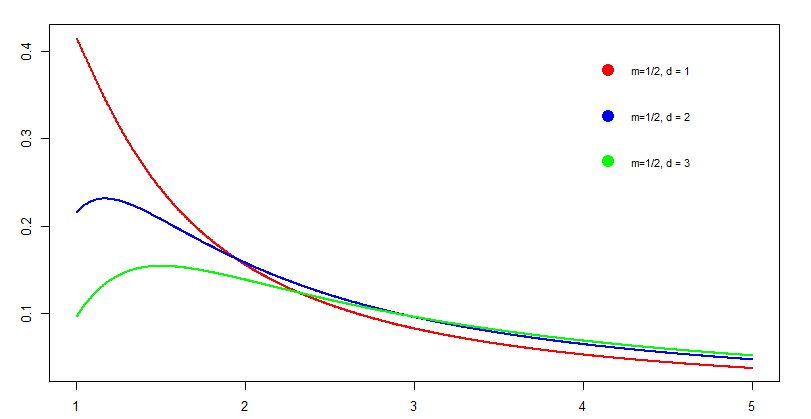
Regresia m2 este cea mai eficienta, aceasta avant atat cel mai mic scor in MSE cat si RMSE

3. **Alegeţi** o repartiţie ***diferită*** de cele studiate la laboratorul sau la cursul de Probabilităţi şi Statistică şi construiţi ȋn două reprezentări alăturate funcţia de masă/densitatea de probabilitate şi respectiv funcţia de repartiţie. Indicaţi proprietăţile pe care le identificaţi la cele două funcţii şi precizaţi la ce este folosită repartiţia respectivă ȋn practică

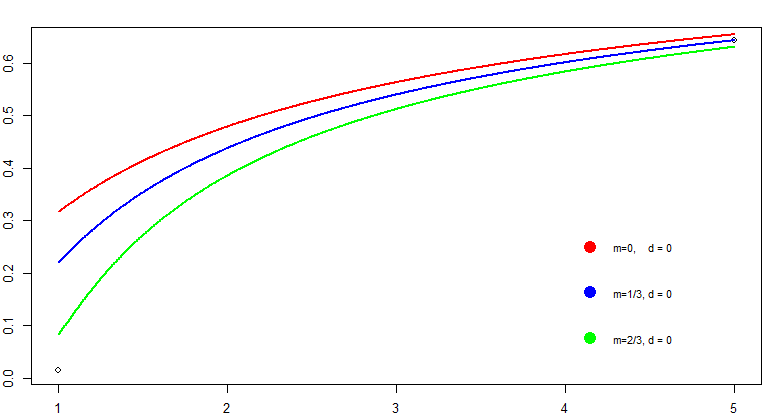
Este necesar packetul ”**rmutil**” pentru a ne folosi de Levy Distribution.



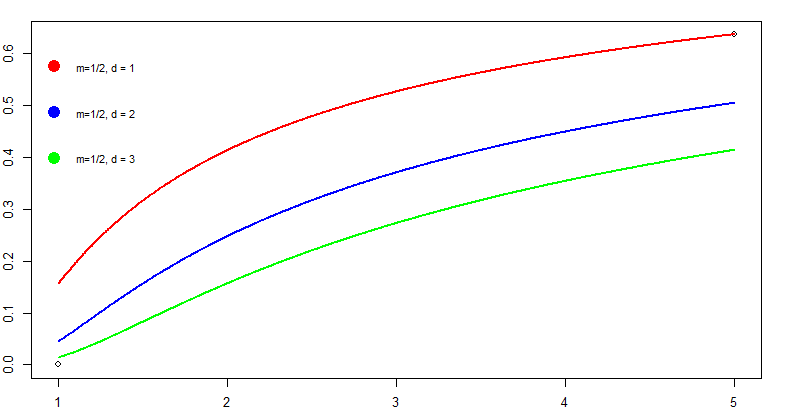
Se poate observa cum creșterea locației face ca punctul de start al graficului să crească, în timp ce ultimele puncte vor începe să tindă asimptotic la 0.



Se poate observa cum mărirea dispersiei scade punctul de start al graficului, cum face punctul de maxim să se deplaseze către dreapta și ca domolește rapiditatea cu care graficul tinde către infinit, astfel încât graficele se inversează.



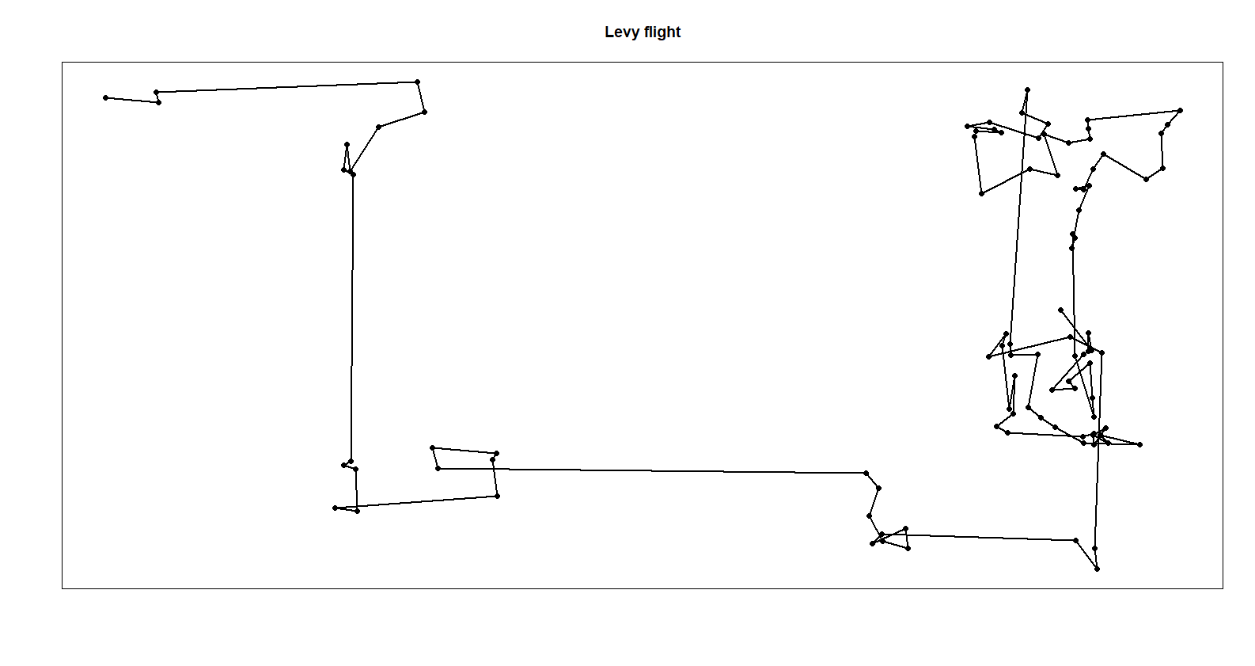
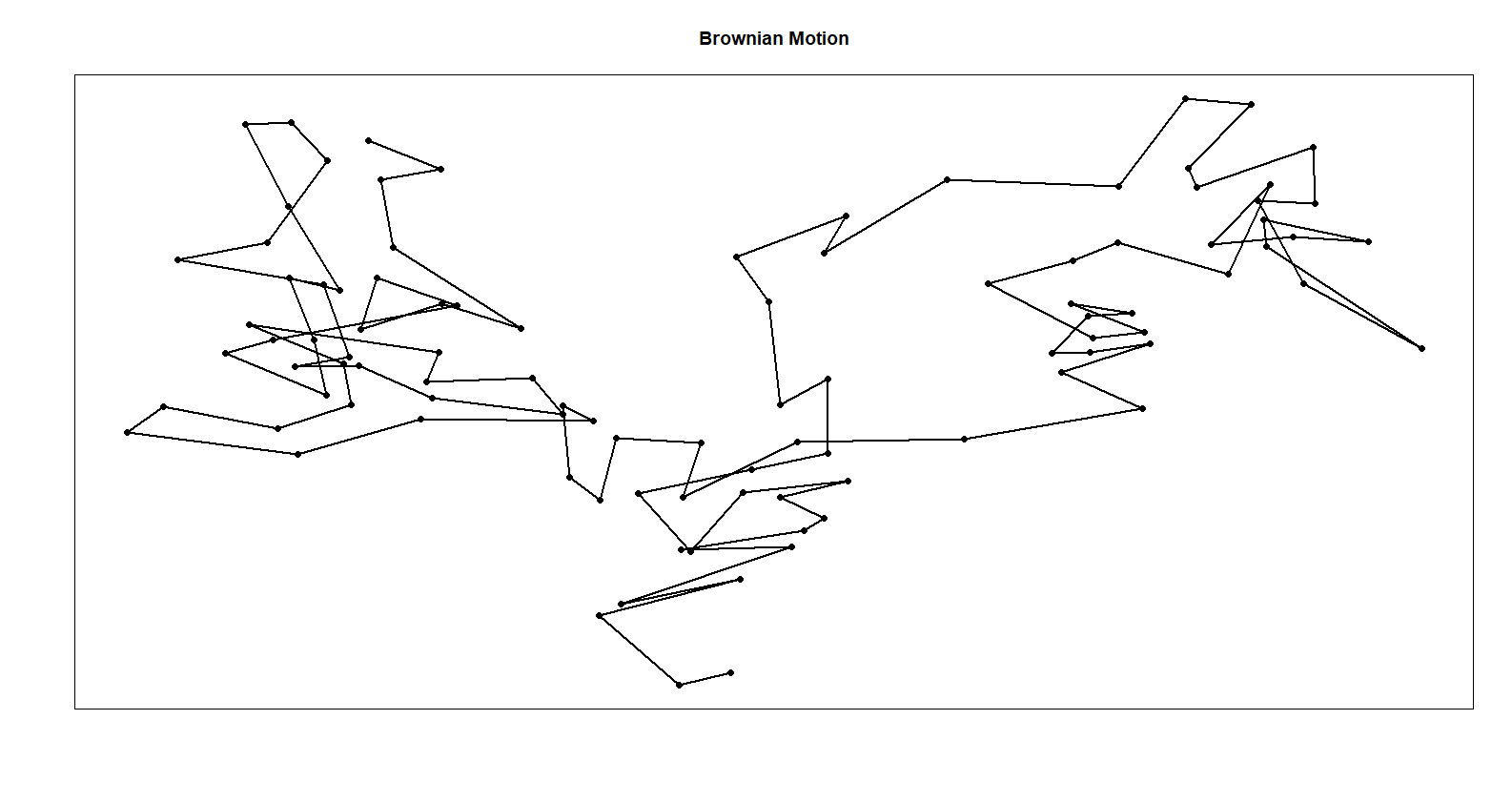
Se poate observa cum creșterea locației face ca punctul de start al graficului să crească, în timp ce ultimele puncte vor începe să tindă asimptotic către același punct.



Se poate observa cum creșterea dispersiei domolește curba graficului și+l face să tindă la un punct, care este mai jos față de graficele cu dispersie mai mica.

Distribuția Levy este folosita pentru a simula epidemii. Asemănătoare cu mișcarea Browniană, doar că aceasta are posibilitatea de a face un salt foarte mare care ar putea însemna infectarea unei țări care nu are nicio legătură cu țările infectate deja, posibil în capătul opus al lumii.

Distribuția Levy poate poate fi folosită și pentru simularea cutremurelor prin simplul fapt că după câțiva pași mici, poate apărea un pas foarte mare (un cutremur foarte mare).



**BONUS:** Fie două variabile aleatoare discrete X şi Y:

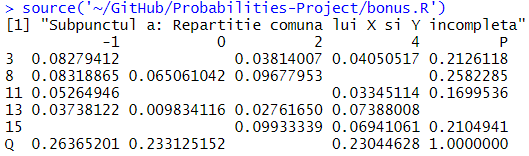
1. Construiţi o funcţie ***frepcomgen*** care primeşte ca parametri ***m*** şi ***n*** şi care generează un tabel cu repartiţia comună a v.a. X şi Y incompletă, dar ȋntr-o formă ȋn care poate fi completată ulterior.

Funcția frepcomgen va genera aleator n \* m elemente (repartitia comuna Pi[i,j]) și va imparti fiecare element la suma tuturor că sa devină o probabilitate.

X are probabilitatiile P[i] = suma probabilitatiilor de pe linia i, Y are Q[j] = suma de pe coloanal j. După ce are o repartiție completă, va scoate elemente după regula:

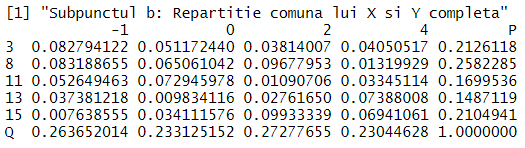
* Se aleg (n+1)\*(m+1) celule din tabelă (inclusiv din coloanal P și linia Q)
* Se elimină fiecare celula [i,j] doar dacă nu a fost deja eliminate (pot apărea duplicate) și daca pana acum s+au eliminat elemente doar de pe linia i sau doar de pe coloana j

Urmând acest algoritm, se poate garanta că exista cel puțin un element care poate fi aflat și care poate duce la aflarea mai multora (ca la sudoku)



1. Construiţi o funcţie ***fcomplrepcom*** care completează repartiţia comună generată la punctul anterior.

Se va completa ca la Sudoku: dacă mai sunt celule necompletate, exista cel puțin una care poate fi completata (acest lucru ne este garantat de modul în care am scos elementele la punctul “a”)



1. c. Având la dispoziţie repartiţia comună a v.a. X şi Y de la punctul b) calculaţi:
2. Cov (5X, -3Y) = -15 \* Cov (X, Y)

Cov (X, Y) se poate calcula

1. P (0<X<3 | Y>2) = 0 dacă P(Y>2) = 0 sau P(0<X<3) = 0

Altfel = P(0<X<3 si Y>2) / P(Y>2)

= sum (tab[xst:xdr, yst:ydr]) / sum (yp[yst:ydr])

1. P(X>6, Y<7) = 0 dacă P(X>6) = 0 sau P(Y<7)

Altfel = P(X>6 si Y<7)

= sum (tab[xst:xdr, yst:ydr])

tab – repartiția comuna de la b = Pi

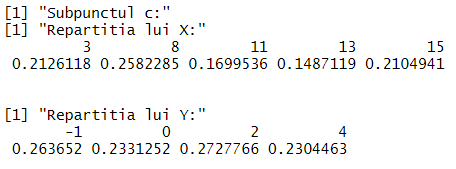
xst – poziția liniei cu cel mai mic id > 0 (excepție ultima = Q)

xdr – poziția liniei cu cel mai mare id < 3 (excepție ultima = Q)

yp – ultima linie din tabel = Q

yst – poziția primei coloane cu cel mai mic id > 2 (excepție ultima = P)

ydr – poziția penultimei coloane (prima înainte de P)



1. Pentru exemplul obţinut la punctul b construiţi două funcţii ***fverind*** şi respectiv ***fvernecor*** cu ajutorul cărora să verificaţi dacă variabilele X şi Y sunt: independente sau necorelate

fverind:

(X Y IND 🡪 cov(X, Y) = 0) ⬄ (cov(X, Y) != 0 🡪 X Y DEP)

Daca cov (x, y) este diferita de 0 🡪 sunt dependente.

Se verifica celulele din Pi daca: Pi[i,j] = P[i] \* Q[j]; daca nu este valabil pentru cel putin o celula 🡪 sunt dependente, altfel sunt IND

fvernecor:

daca cov(X, Y) = 0 atunci sunt NEC, altfel sunt corelate.

