

Funcții care procesează liste

all(list)

Returnează 1 dacă *list* conține doar elemente nenule

any(list)

Returnează 1 dacă *list* conține cel puțin un element nenul.

cumsum(list)

Calculează lista sumelor parțiale ale listei date ($S_k = l_1 + \dots + l_k$).
`cumsum([1,2,3]) => [1 3 6]`

find(list)

Returnează lista indicilor elementelor nenule din *list*

intersect(a,b)

Returnează lista elementelor comune listelor *a* și *b*

ismember(a,s)

Verifică existența lui *a* în lista *s*. Dacă *a* este listă, se verifică dacă fiecare element al lui *a* se regăsește în *s*
`ismember([1,10,3], 0:9) => [0 1 0]`

length(list)

Returnează lungimea listei

max(list)

Returnează maximul din *list*

ones(m)

ones(m,n)

Echivalent cu `repelem(1,m), repelem(1,m,n)`

prod(list)

Returnează produsul elementelor din *list*

repelem(x,m)

Returnează un vector de lungime *m* care îl conține numai pe *x*

repelem(x,m,n)

Returnează o matrice de lungime $m \times n$ care îl conține numai pe *x*

repmat(A,m)

repmat(A,m,n)

Creează o matrice $m \times n$ prin duplicarea matricei *A*. Când *n* nu este specificat, se consideră $m = n$.
`repmat([5], 1, 3) => [5,5,5]`

sort(list)

Sortează *list*

sum(list)

Returnează suma elementelor din *list*

union(a,b)

Returnează reuniunea listelor *a* și *b*

unique(list)

Returnează lista sortată a elementelor din *list*, fiecare luat câte o singură dată

zeros(m)

zeros(m,n)

Echivalent cu `repelem(0,m), repelem(0,m,n)`

[a,b]

Concatenare liste pe următoarea coloană
`[1,2] => [1 2]`
`[1, [2,3]] => [1 2 3]`

[a;b]

Concatenare liste pe următoarea linie
`[1, 2; 3, 4] => 1 2`
`3 4`

a .+ b, a .- b, a .* b, a ./ b

Efectuează operații aritmetice pe liste, operând pe rând elementele două câte două.
`[1, 2] .+ [3, 4] => [4, 6]`

a + s, a - s, a * s, a / s

Operații aritmetice între o listă și un scalar, se aplică pe rând între fiecare element al listei și scalar
`[1,2] * 4 => [4,8]`

a:b, a:c:b

Generează o listă începând cu *a* de elemente în progresie aritmetică până la *b*, cu pasul 1, sau *c*, când e specificat
`1:3:11 => [1 4 7 10]; 2:5 => [2 3 4 5]`

size(list)

Returnează mărimea unei liste văzute ca o matrice 2D.
`size([1 2 3;4 5 6]) => [2 3]`

Funcții care calculează chestii exacte

linspace(a,b)

linspace(a,b,n)

Returnează o listă de *n* numere de la *a* la *b* inclusiv, egal depărtate unul de altul. By default, $n = 100$

nchoosek(n,k)

Calculează C_n^k

nchoosek(list,k)

Generează toate combinațiile de *k* elemente cu elementele din *list*
`nchoosek([1,2,3],2) => [1 2;1 3;2 3]`

perms(list)

Generează toate permutările elementelor din *list*

Funcții care generează chestii random

rand()

Generează un număr random subunitar

rand(m,n)

Generează o matrice $m \times n$ de numere subunitare

randi(k)

Generează un întreg random de la 1 la k

randi(k,m,n)

Generează o matrice $m \times n$ de numere random de la 1 la k

`randi(5,1,3) => [5,4,2]`

randperm(n)

Generează o permutare random de ordinul n

`randperm(5) => [2 4 1 3 5]`

randsample(n,k)

Alege k numere de la 1 la n

`randsample(10,3) => [10,5,8]`

randsample(list, k)

Alege k numere din *list*, $k \leq \text{len}(\text{list})$

`randsample([1,2,3],2) => [3,1]`

Generare random conform unei distribuții

binornd(n,p)

binornd(n,p,r)

binornd(n,p,r,c)

Returnează o matrice 1×1 , $r \times r$ sau $r \times c$ de numere generate conform distribuției binomiale de parametri n și p .

exprnd(lambda)

exprnd(lambda,r)

exprnd(lambda,r,c)

Returnează o matrice 1×1 , $r \times r$ sau $r \times c$ de numere generate conform distribuției exponențiale de parametru *lambda*.

geornd(p)

geornd(p,r)

geornd(p,r,c)

Returnează o matrice 1×1 , $r \times r$ sau $r \times c$ de numere generate conform distribuției geometrice de probabilitate p .

hygernd(t,m,n)

hygernd(t,m,n,r)

hygernd(t,m,n,r,c)

Returnează o matrice 1×1 , $r \times r$ sau $r \times c$ de numere generate conform distribuției hipergeometrice de parametri t, m, n .

unidrnd(n)

unidrnd(n,r)

unidrnd(n,r,c)

Returnează o matrice 1×1 , $r \times r$ sau $r \times c$ de numere generate conform distribuției uniforme pe numerele naturale de la 1 la n .

unifrnd(a,b)

unifrnd(a,b,r)

unifrnd(a,b,r,c)

Returnează o matrice 1×1 , $r \times r$ sau $r \times c$ de numere generate conform distribuției uniforme pe numerele reale din intervalul $[a, b]$.

Probabilitatea teoretică la distribuții

binopdf(k,n,p)

Valoarea în k a funcției de densitate pentru distribuția binomială de parametri n și p .

$$\text{binopdf}(k, n, p) = C_n^k p^k (1-p)^{n-k}$$

n = numărul de încercări

p = probabilitatea de succes

exppdf(x,m)

Valoarea în x a funcției de densitate pentru distribuția exponențială cu media m .

$$\text{exppdf}(x, m) = \frac{1}{m} e^{-\frac{x}{m}}$$

geopdf(k,p)

Valoarea în k a funcției de densitate pentru distribuția geometrică cu probabilitatea p .

$$\text{geopdf}(k, p) = p(1-p)^k$$

k = numărul de eșecuri până la primul succes

hygepdf(x,m,k,n)

Valoarea în x a funcției de densitate pentru distribuția hipergeometrică de parametri m, k, n .

$$\text{hygepdf}(x, m, k, n) = \frac{C_k^x \cdot C_{m-k}^{n-x}}{C_m^n}$$

n = numărul total de entități

k = numărul de entități cu o proprietate P ,

$k \leq n$

m = numărul de extrageri

$hygepdf(x, m, k, n)$ = probabilitatea de a obține x entități cu proprietatea P din m extrageri.

unidpdf(k,n)

Valoarea în k a funcției de densitate pentru distribuția uniformă discretă de parametru n .

$$unidpdf(k, n) = \frac{1}{n}$$

unifpdf(x,a,b)

Valoarea în k a funcției de densitate pentru distribuția uniform de parametri a, b .

$$unifpdf(k, a, b) = \frac{1}{b - a}$$

Chestii ceva cu funcții de densitate

mean(list)

Calculează media aritmetică a elementelor din *list*

`mean([1,2,4]) => 2.3333`

std(list)

Calculează deviația standard a elementelor din *list*

$$std(x_1 \dots x_n) = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - mean(x))^2}$$

Ce or mai fi și astea

binocdf(x,n,p)

expcdf(x,m)

geocdf(x,p)

hygecdf(x,m,k,n)

unidcdf(x,n)

unifcdf(x,a,b)

Calculează valoarea distribuției cumulative a modelului de distribuție corespunzător.

$$\mathcal{D}_{cdf}(x, params) = \sum_{y \leq x} \mathcal{D}_{pdf}(y, params)$$

unde \mathcal{D} este un model de distribuție (*binocdf*, *exp*, *geo* etc.)

Matimatici

exp(x)

integral(f,a,b)

ceil(x)

floor(x)

abs(x)

mod(a,b)

sqrt(x)

log(x)

sin(x)

cos(x)

pi

Când sunt apelate pe liste, se apelează funcția pe fiecare element al listei.

`cos([a b c]) => [cos(a) cos(b) cos(c)]`

pdist(x)

Calculează distanțele două câte două dintre punctele ale căror coordonate se află pe liniile matricei x .

`pdist([0 5; 1 5; 3 5; 4 5]) = [1 3 4 2 3 1]`

Explicație: punctele sunt: $A(0,5)$, $B(1,5)$, $C(3,5)$, $D(4,5)$. Distanțele sunt:

$AB = 1$, $AC = 3$, $AD = 4$, $BC = 2$, $BD = 3$, $CD = 1$

Desenare

clf; grid on; hold on;

Face un fel de clear screen, naiba știe.

hist(y)

hist(y,x)

Histogramă pe lista y . Numără de câte ori apare fiecare element în lista y . Când apelul nu se află în urma unei atribuirii, se va afișa figura. Altfel, se returnează două liste $[c, h]$. Lista c conține cantitățile în fiecare punct al histogramei. Lista h , Dumnezeu știe. Când x este specificat, acesta reprezintă numărul de unități în care este împărțită histograma.

bar(x, y, 'FaceColor', color)

Grafic cu bare între 2 vectori x și y , cu însemnătatea că lui x_i îi corespunde valoarea y_i . *color* poate fi un caracter.

figure(n)

Creează o fereastră nouă cu id-ul n , sau o accesează dacă este deja deschisă.