Calcul des probabilités
Expressions de lois de probabilité

Michel DE LARA

October 10, 2017

Contents

Écrire la suite d'instructions suivante dans un fichier `nom_du_fichier.sci`, puis les exécuter avec la commande `getf("nom_du_fichier.sci","c");`

```
function introduction()
  // à mettre dans un fichier nom_du_fichier.sci
  // et utiliser avec exec("nom_du_fichier.sci","c");
endfunction
```

```
function H=Heavyside(x)
  // Fonction de Heavyside
  H=max(sign(x),0);
endfunction
```

1 Lois à densité

1.1 Loi normale

```
function Ga=Gauss(x,mu,sigma)
  // loi de Gauss reelle
  // sigma est ici l'écart-type
  //--------------------
  pi=3.1415927;
  theta=(x-mu)/sigma;
  Ga=1/(sigma*sqrt(2*pi))*exp(-theta^2/2);
endfunction
```

```
function Ga=d_normal(x,mu,Sigma)
  // d_normal : loi de Gauss
  // mu : vecteur colonne (n,1)
  // Sigma: la matrice de variance
  // x: matrice formee de m vecteurs colonnes (n,1);
  [m,n]=size(x);
  Ga=[];
  c=(1/(2*pi)^(m/2))*1/sqrt(det(Sigma));
  for j=1:n do
    Ga=[Ga,c*exp(-(x(:,j)-mu)'*inv(Sigma)*(x(:,j)-mu)/2)];
  end;
endfunction
```

1.2 Loi uniforme

```
function U=d_uniform(x,a,b)
  // d_uniform
  U=Heavyside(x-a) .*Heavyside(b-x) ./(b-a);
endfunction
```

1.3 Loi exponentielle

```
function E=d_exponential(x,lambda)
  // d_exponential
  E=Heavyside(x) .*lambda .*exp(-lambda .*x);
endfunction
```

1.4 Loi de Weibull

```
function W=d_weibull(x,a,alpha)
  // d_weibull
  // d'apr\`es /u/cergrene/0/jpc/Scilab/STIXBOX/
  if or(mtlb_any(a <= 0 | alpha <= 0)) then
    error('Parameter a or alpha is nonpositive');
  end
  I=mtlb_find(x <= 0);
  W=a .*x .^(alpha-1) .*exp(-a .*x .^(alpha) ./alpha);
  W(I)=0*I;
endfunction
```

1.5 Loi de Cauchy

```
function C=d_cauchy(x,a)
  //d_cauchy
  pi=3.1415927;
  C=1/pi .*a ./(a^2+x^2);
endfunction
```

1.6 Loi log-normale

```
function LN=d_lognormal(x,mu,sigma)
  // d_lognormal
  pi=3.1415927;
  //    y=max(x,0.00000001);
  theta=(log(x)-mu) ./sigma;
  LN=1 ./(x .*sigma .*sqrt(2*pi)) .*exp(-theta .^2 ./2);
  I=mtlb_find(x <= 0);
  LN(I)=0*I;
endfunction
```

1.7 Loi beta

```
function d=d_beta(x,a,b)
  // d_beta
  //d=[];
  if or(mtlb_any(a <= 0 | b <= 0)) then
    error('Parameter a or b is nonpositive');
  end
  I=mtlb_find(x <= 0 | x >= 1);
  d=real(x .^(a-1) .*(1-x) .^(b-1) ./beta(a,b));
  d(I)=0*I;
endfunction
```

1.8 Loi gamma

```
function f=d_gamma(x,a,b)
  // d_gamma
  f=[];
  if or(mtlb_any(a <= 0 | b <= 0)) then
    error('Parameter a or b is nonpositive');
  end
  f=real(x .^(a-1) .*exp(-x ./b) ./(gamma(a) .*b .^a));
  I0=mtlb_find(x <= 0);
  %v=size(I0)
  f(I0)=real(zeros(%v(1),%v(2)));
endfunction
```

1.9 Loi du chi-deux

```
function X=d_chisquare(x,n)
  // d_chisquare
  X=d_gamma(x/2,n*0.5,1)/2;
endfunction
```

1.10 Loi de Student

```
function S=d_student(x,n)
  // d_student
  pi=3.1415927;
  S=gamma((n+1)/2)/(gamma(n/2)*sqrt(n*pi)) .*((1+x .^2) ./n) .^(-(n+1) ./2);
endfunction
```

1.11 Loi de Fisher

```
function F=d_fisher(x,n,m)
  // d_fisher
  y=[0*x(x <= 0),x(x > 0)];
  F=Heavyside(x) .*gamma((n+m)/2) .*n^(n/2) .*m^(m/2) ./(gamma(n/2) .*gamma(m/2)) .* ...
    (y .^(n/2-1)) ./((m+n .*y) .^((n+m)/2));
  //    F=real(F);
  //    c = m ./ n; xx = x ./ (x+c);
  //    f = d_beta(xx,n/2,m/2);      F = f ./ ((x+c).^2)*c;
endfunction
```

2 Lois discrètes

2.1 Loi binômiale

```
function zt=d_binomial(i,N,p)
  // d_binomial
  //-------------------------------
  function y=fact(n) y=prod(1:n);endfunction
  function z=C(N,n) z=fact(N)/(fact(n)*fact(N-n));endfunction
  zt=[];
  for j=i do zt=[zt,C(N,j)*p^j*(1-p)^(N-j)];end
  I=mtlb_find(i < 0 | i > N);
  zt(I)=0*I;
endfunction
```

2.2 Loi géométrique

```
function zt=d_geometric(i,p)
  // d_geometric
  //-------------------------------
  zt=[];
  for j=i do zt=[zt,p*(1-p)^(j-1)];end
  I=mtlb_find(i <= 0);
  zt(I)=0*I;
endfunction
```

2.3 Loi hypergéométrique

```
function zt=d_hypergeometric(i,n,N1,N2)
  // d_hypergeometric
  //-------------------------------
```

```
  function y=fact(n) y=prod(1:n);endfunction
  function z=C(N,n) z=fact(N)/(fact(n)*fact(N-n));endfunction
  zt=[];
  for j=i do zt=[zt,C(N1,j)*C(N2,n-j)/C(N1+N2,n)];end
  I=mtlb_find(i < max(n-N2,0) | i > min(N1,n));
  zt(I)=0*I;
endfunction
```

2.4 Loi binômiale négative

```
function zt=d_negativebinomial(i,r,p)
  // d_negativebinomial
  //-------------------------------
  function y=fact(n) y=prod(1:n);endfunction
  function z=C(N,n) z=fact(N)/(fact(n)*fact(N-n));endfunction
  zt=[];
  for j=i do zt=[zt,C(j+r-1,r-1)*p^r*(1-p)^j];end
  I=mtlb_find(i < 0);
  zt(I)=0*I;
endfunction
```

2.5 Loi de Pascal

```
function zt=Pascal(i,p,r)
  //-------------------------------
  function y=fact(n) y=prod(1:n);endfunction
  function z=C(N,n) z=fact(N)/(fact(n)*fact(N-n));endfunction
  zt=[];
  for j=i do zt=[zt,C(j-1,r-1)*p^r*(1-p)^(j-r)];end
  I=mtlb_find(i < r);
  zt(I)=0*I;
endfunction
```

2.6 Loi de Poisson

```
function zt=d_poisson(i,lambda)
  // d_poisson
  //-------------------------------
  function y=fact(n) y=prod(1:n);endfunction
  zt=[];
  for j=i do zt=[zt,exp(-lambda)*lambda^j/fact(j)];end
  I=mtlb_find(i < 0);
```

```
  zt(I)=0*I;
endfunction
```

2.7 Autres lois

```
function zt=Malchance(i)
  // loi de persistance de malchance
  //I = mtlb_find(i>=2);
  zt=[];
  for j=i do zt=[zt,1/(j-1)-(1/j)];end
endfunction

function zt=Renversement(i)
  // loi du temps de premier renversement
  //I = mtlb_find(i>=2);
  function y=fact(n) y=prod(1:n);endfunction
  zt=[];
  for j=i do zt=[zt,1/fact(j-1)-1/fact(j)];end
endfunction
```