Loi de Bernoulli et loi binomiale, cours, première S

1 Loi de Bernoulli

Définition :
Soit p un nombre réel tel que $p \in [0;1]$. Soit X une variable aléatoire. On dit que X suit une loi de Bernoulli de paramètre p si :

- X prend pour seules valeurs 1 (« succès ») et 0 (« échec ») ;
- $P(X=1)=p$ et $P(X=0)=1-p$.

Propriété :
Soit X une variable aléatoire qui suit la loi de Bernoulli de paramètre p.

- $E(X)=p$;
- $V(X)=p(1-p)$.

Preuve :
D'une part, $E(X)=0\times P(X=0)+1\times P(X=1)=0\times(1-p)+1\times p=p$.
En outre, $V(X)=P(X=0)\times(0-E(X))^2+P(X=1)\times(1-E(X))^2=(1-p)p^2+p(1-p)^2=p(1-p)(p+1-p)=p(1-p)$

Algorithmique :
Algorithme de simulation d'une épreuve de Bernoulli de paramètre p.

Données : p : nombre décimal entre 0 et 1 ;
Début traitement
  t prend une valeur aléatoire décimale entre 0 inclus et 1 exclu ;
  si $t<p$ **alors**
    Afficher "Succès" ;
  fin
  sinon
    Afficher "Échec" ;
  fin
Fin

Exemple :

<table>
<tr>
<td>

TI :

Prompt P
NbreAleatoire() $\rhd T$
If $T < P$
Then
Disp "SUCCES"
Else
Disp "ECHEC"

</td>
<td>

Casio :

$"P"? \to P$
Rand# $\to T$
If $T < P$
Then "SUCCES"
Else "ECHEC"

</td>
</tr>
<tr>
<td>

XCas :

```
saisir("Entrer p : ",p);
t:=alea(0,1);
si (t<p) alors
  afficher("Succès");
sinon
  afficher("Echec");
fsi;
```

</td>
<td>

Python :

```
from random import*
p=float(raw_input("Entrer p : "))
t=random()
if (t<p):
  print("Succès")
else:
  print("Echec")
```

</td>
</tr>
</table>

2 Schéma de Bernoulli

Définition :

Deux expériences sont dites indépendantes si le résultat de l'une n'influence pas le résultat de l'autre.

Exemple :
il y a indépendance lorsqu'on lance deux fois de suite une pièce de monnaie.

Définition :

La répétition d'une expérience aléatoire suivant une loi de Bernoulli de paramètre p, ceci n fois de manière indépendante, constitue *un schéma de Bernoulli* de paramètres n et p.

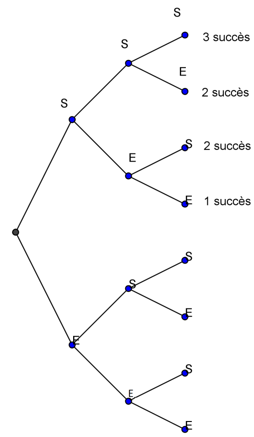
Propriété :

Dans un schéma de Bernoulli, la probabilité d'une liste de résultats est le produit des probabilités de chaque résultat

Exemple de savoir faire :

- **[Construire un arbre représentant un schéma de Bernoulli de paramètres donnés]**
  On contrôle la qualité d'un produit sur une chaîne de production. On prélève 3 produits au

hasard. On suppose que les prélèvements sont indépendants. Statistiquement, chaque produit a une probabilité $p = 0,05$ d'être défectueux.
On a donc un schéma de Bernoulli de paramètres $p = 0,05$ et $n = 3$.



- **[Calculer la probabilité d'un événement représenté par un chemin sur un arbre pondéré]**
  Sur l'arbre ci-dessus représentant un schéma de Bernoulli de paramètres $n = 3$ et $p = 0,05$, la probabilité d'avoir les deux premières expériences qui donnent un succès et la dernière qui donne un échec est $P(SS\bar{S}) = 0,05^2 \times 0,95 \approx 0,002$
  soit 0,2 % de chances d'avoir deux produits défectueux sur les trois prélevés.

3 Loi binomiale

Définition et propriété :
Soient n et k deux entiers naturels avec $k \leq n$. On note $\binom{n}{k}$ (on dit « k sous n ») le nombre de manières d'obtenir k succès et $n - k$ échecs pour n répétitions indépendantes de la même expérience de Bernoulli.

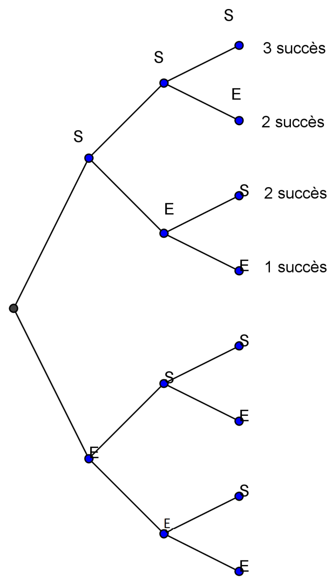
Remarque hors programme mais culturelle :
On a

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

avec $n! = n \times (n-1) \times (n-2) \times ... \times 2 \times 1$.

Preuve :
Admise

Exemple :



$\binom{3}{3} = 1$: il y a une seule manière d'obtenir 3 succès lors de la répétition de 3 épreuves identiques indépendantes.
$\binom{3}{2} = 3$: il y a trois manières d'obtenir 2 succès et un échec lors de la répétition de 3 épreuves identiques indépendantes (SSE ; SES ; ESS).

Définition :
On considère une épreuve de Bernoulli de paramètre $p \in [0;1]$. On répète n fois ($n \geq 1$) cette expérience indépendamment et on note X la variable aléatoire qui compte le nombre de succés. On dit alors que la variable aléatoire X suit une loi *binomiale* de paramètres n et p et on note $X \sim \mathcal{B}(n;p)$.

Propriété :
Si X est une variable aléatoire qui suit une loi binomiale de paramètres n et p, alors la probabilité d'obtenir k succès avec $k \in \{0;1;2;...;n\}$ est $P(X=k) = \binom{n}{k}p^k(1-p)^{n-k}$.

Preuve :
Il y a $\binom{n}{k}$ manières d'obtenir k succès dans n répétitions d'expériences identiques et indépendantes. La probabilité de chacune de ces événements qui sont évidemment incompatibles est $p^k(1-p)^{n-k}$ d'où le résultat.

Exemple de savoir faire :
[Calculer la probabilité de $P(X=k)$ où X suit une loi binomiale à partir d'un arbre pondéré]
On considère le problème précédent de test des produits d'une chaîne de production. Les prélèvements étant supposés indépendants les uns des autres, l'expérience constitue un schéma de Bernoulli de paramètres $n=3$ et $p=0,05$. La variable aléatoire X qui compte le nombre de succès suit la loi binomiale de paramètres $n=3$ et $p=0,05$.

On a $P(X=2)=P(SS\bar{S}\cap S\bar{S}S\cap\bar{S}SS)$
car trois chemins permettent d'obtenir deux succès c'est à dire deux objets défectueux.
D'où $P(X=2)=P(SS\bar{S})+P(S\bar{S}S)+P(\bar{S}SS)$
donc $P(X=2)=0,05^2\times 0,95+0,95\times 0,05\times 0,95+0,95\times 0,05^2=3\times 0,05^2\times 0,95=0,007$
soit une probabilité très faible de 0,007 d'avoir deux produits défectueux.

Calcul pratique de $P(X=k)$ et $P(X\leq k)$:
Soit X une variable aléatoire de paramètres n et p. Pour k allant de 0 à n, pour calculer $P(X=k)$ ou $P(X\leq k)$, on utilise une calculatrice :

- Sur Texas instrument : aller dans le menu $\boxed{\text{2nd}}\boxed{\text{distrib}}$, choisir $\boxed{\text{binomFdp}}$ et taper $\boxed{\text{n}}\boxed{,}\boxed{\text{p}}\boxed{,}\boxed{\text{k}}\boxed{)}$ pour calculer $P(X=k)$ et choisir $\boxed{\text{binomFRép}}$ et taper $\boxed{\text{n}}\boxed{,}\boxed{\text{p}}\boxed{,}\boxed{\text{k}}\boxed{)}$ pour calculer $P(X\leq k)$.
- Sur Casio : aller dans le menu $\boxed{\text{STAT}}$ puis $\boxed{\text{DIST}}$ puis $\boxed{\text{BINM}}$. Sélectionner alors $\boxed{\text{Bpd}}$ puis $\boxed{\text{Var}}$ pour variable, puis entrer alors k dans la ligne « x », n dans la ligne « numtrial » et p dans la ligne « p » puis aller sur « execute » pour valider et calculer ainsi $P(X=k)$. Pour le calcul de $P(X\leq k)$, on utilisera $\boxed{\text{Bcd}}$ au lieu de $\boxed{\text{Bpd}}$.

Exemple de savoir faire :
[Calculer $P(X=k)$ ou $P(X\leq k)$ où X suit une loi binomiale à l'aide d'une calculatrice]
On considère une variable aléatoire X suivant la loi binomiale de paramètres $n=4$ et $p=0,4$.
Sur TI, la probabilité $P(X\leq 3)$ est donnée par `binomFrép(4,0.4,3)`.

Remarques :

- On a $P(X<k)=P(X\leq k-1)$
- pour calculer $P(X>k)$, on calcule $1-P(X\leq k)$.

Propriété :
Soit X une variable aléatoire qui suit une loi binomiale de paramètres n et p. Alors :

- $E(X)=np$;
- $V(X)=np(1-p)$

Preuve :
Admise

Algorithmique :
Algorithme de simulation d'une loi binomiale de paramètres n et p.

Exemple :

Données : p : nombre décimal entre 0 et 1 ; n : entier naturel ;
Début traitement
  c prend la valeur 0 ;
  pour k *de 1 jusque* n **faire**
    t prend une valeur aléatoire décimale entre 0 inclus et 1 exclu ;
    si $t < p$ **alors**
      c prend la valeur de $c + 1$;
    fin
  fin
Fin
Sorties : c

TI : Prompt N Prompt P $0 \triangleright C$ For(K,1,N) $NbreAleatoire() \triangleright T$ If $T < P$ Then $C + 1 \triangleright C$ End End Disp C	**Casio :** $"P"? \to P$ $"N"? \to N$ $0 \to C$ For $1 \to K$ To N Step 1 Rand#$\to T$ If $T < P$ Then $C + 1 \to C$ IfEnd Next "$C=$" :C

XCas :
`saisir("Entrer p : ",p);`
`saisir("Entrer n : ",n);`
`c:=0;`
`pour k de 1 jusque n faire`
`  t:=alea(0,1);`
`  si t<p alors`
`    c:=c+1;`
`  fsi;`
`fpour"`
`afficher("Nombre de succès : "+c);`
	Python :
	`from random import*`
	`p=float(raw_input("Entrer p : "))`
	`n=int(raw_input("Entrer n : "))`
	`c=0`
	`for k in range(1,n+1):`
	`  t=random()`
	`  if t<p:`
	`    c=c+1`
	`print("Nombre de succès",c)`

Algorithmique :

Algorithme de simulation d'obtention de N échantillons de lois binomiales de paramètres n et p.

Données : n, N : nombres entiers ;p : nombre décimal entre 0 et 1 ;
Début traitement
  s est une liste vide ;
  pour *k de 0 jusque n* **faire**
    $s[k]$ prend la valeur 0 ;
  fin
  pour *m de 1 jusque N* **faire**
    c prend la valeur 0 ;
    pour *k de 1 jusque n* **faire**
      t prend une valeur aléatoire décimale entre 0 inclus et 1 exclu ;
      si $t < p$ **alors**
        c prend la valeur de $c + 1$;
      fin
    fin
    $s[c]$ prend la valeur de $s[c] + 1$;
  fin
  Afficher s ;
Fin

Exemple :

XCas :

```
saisir("Entrer p : ",p);
saisir("Entrer n : ",n);
saisir("Entrer N : ",N);
s:=[];
pour k de 0 jusque n faire
  s[k]:=0;
fpour;
pour m de 1 jusque N faire
  c:=0;
  pour k de 1 jusque n faire
    t:=alea(0,1);
    si t<p alors
      c:=c+1;
    fsi;
  fpour;
  s[c]:=s[c]+1;
fpour;
pour k de 0 jusque n faire
  afficher("Avec "+k+" succès : "+s[k]);
fpour;
```

Python :

```
from random import*
p=float(raw_input("Entrer p : "))
n=int(raw_input("Entrer n : "))
N=int(raw_input("Entrer N : "))
s=[]
for k in range(0,n+1):
  s.append(0)
for m in range(1,N+1):
  c=0
  for k in range(1,n+1):
    t=random()
    if t<p:
      c=c+1
  s[c]=s[c]+1
for k in range(0,n+1):
print "Nombre avec "+k+" succès : "+s[k]
```

4 Coefficients binomiaux

Propriétés :

- $\binom{n}{k} = \binom{n}{n-k}$;
- $\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}$

Triangle de Pascal :
Le triangle de Pascal, du nom de Blaise Pascal, mathématicien français du XVII^{e} siècle qui le« redécouvra » en Occident (car il était connu avant en Orient et au Moyen-Orient) est une disposition permettant de visualiser et de calculer les coefficients binomiaux et qui s'appuie sur la formule précédente.

$\binom{0}{0} = 1$					
$\binom{1}{0} = 1$	$\binom{1}{1} = 1$				
$\binom{2}{0} = 1$	$\binom{2}{1} = 2$	1			
$\binom{3}{0} = 1$	$\binom{3}{1} = 3$	3	1		
1	4	6	4	1	
1	5	10	10	5	1

Explication de la construction : le nombre de la ligne n et de la colonne k est le coefficient binomial $\binom{n-1}{k-1}$. Il est obtenu en ajoutant le nombre situé au dessus (ligne $n-1$ et colonne k) au nombre de la colonne et de la ligne précédente (ligne $n-1$ et colonne $k-1$).
Par exemple, $\binom{3}{1} = 3$ est la somme de $\binom{2}{1} = 2$ et de $\binom{2}{0} = 1$.

1					
1	1				
1	2	1			
1	3	3	1		
1	4	6	4	1	
1	5	10	10	5	1

Exemple de programmation du triangle de Pascal en langage python :

```python
def trianglePascal(n):
        L=[1]
        L=L+[0 for k in range(n)]
        t=[L]
        for k in range(1,n+1):
                L=[1]
                for l in range(1,k+1):
                        L.append(t[k-1][l]+t[k-1][l-1])
                L=L+[0 for l in range(k,n)]
                t=t+[L]
        return t
```