Lois usuelles des VAR sur un EPF

Rappels

- $(\Omega, \mathscr{P}(\Omega), P)$ est un espace probabilisé fini.
- La variable est notée X.
- Si $p \in [0,1]$ on pose $q = 1 - p$.

I Loi certaine $\mathscr{C}(\lambda)$

Description. C'est la loi du : pas d'aléa.

Paramètre. λ : réel.

Espace image. $X(\Omega) = \{\lambda\}$

Loi.

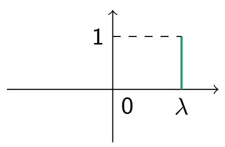
$x =$	λ
$P(X = x)$	1

Espérance. $E(X) = \lambda$.

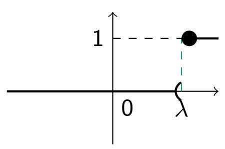
Variance. $V(X) = 0$.

Exemple. Une urne contient N boules blanches. On tire λ boules simultanément, et X est la variable aléatoire qui compte le nombre de boules blanches obtenues.

Peigne.



Fonction de répartition



II Loi uniforme $\mathscr{U}(n)$

Description. C'est la loi du pur hasard.

Paramètre. n : entier strictement positif.

Espace image. $X(\Omega) = \{1, 2, \dots, n\}$.
☠ Attention : 0 n'est pas dans $X(\Omega)$. La VAR qui prend équiprobablement les valeurs de $\{0 \dots n\}$ ne suit pas la loi appelée $\mathscr{U}(n)$.

Loi.

k=	1	2	…	n
$P(X = k)$	$\frac{1}{n}$	$\frac{1}{n}$	…	$\frac{1}{n}$

Espérance. $E(X) = \frac{n+1}{2}$

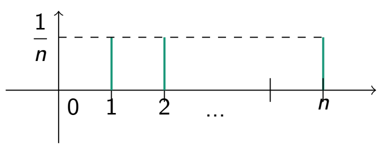
Variance. $V(X) = \frac{n^2-1}{12}$ (pas à connaître).

Exemple. Une urne contient n boules numérotées de 1 à n. On tire une boule au hasard et on note X la VAR égale au numéro de la boule tirée.

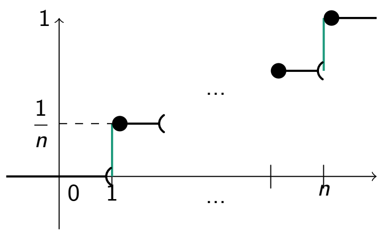
Simulation. Code Python ▼

```python
from random import randint
n = 10 # loi U(10)
randint(1,10)
```

Peigne.



Fonction de répartition.



III Loi de Bernoulli $\mathscr{B}(p)$

Description. C'est la loi du ça passe ou ça casse.

Paramètre. $p \in]0,1[$.

Espace image. $X(\Omega) = \{0, 1\}$.
Conventionnellement, 0 code l'issue «échec» et 1 code l'issue «succès». L'expérience décrite par X est celle d'une épreuve à deux issues (ca passe ou ça casse).

Loi.

$k =$	0	1
$P(X = k)$	$1 - p$	p

Espérance. $E(X) = p$.

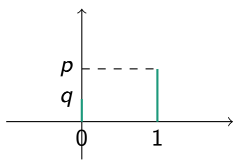
Variance. $V(X) = pq$.

Exemple. Une urne contient n boules, dont une proportion p est blanche. On tire une boule au hasard, et X est la variable aléatoire qui vaut 1 si la boule tirée est blanche, et 0 sinon.

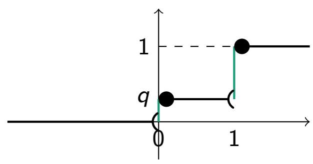
Simulation. Code Python ▼

```python
import random as rd
p = 0.3 # Loi  B(p)
int(rd.random()<p)
```

Peigne.



Fonction de répartition



IV Loi bimomiale $\mathscr{B}(n, p)$

Description. Loi du nombre de succés lors de la répétition de N épreuves de Bernoulli de même paramètre et **mutuellement indépendantes**.

Paramètres. $N > 0$: entier, $p \in]0, 1[$.

Espace image. $X(\Omega) = \{0, \dots n\}$. Attention, 0 est dans l'espace image.

Loi. $\forall k \in \{0 \dots n\}$

$$P(X = k) = \binom{n}{k} p^k q^{n-k}$$

Remarque. La loi est particulièrement facile à calculer dans le cas où $p = q = \dfrac{1}{2}$ puisqu'elle s'obtient en divisant les coefficients binomiaux par 2^n. Ci-dessous, la loi $\mathscr{B}\left(5, \dfrac{1}{2}\right)$:

$k =$	0	1	2	3	4	5
$P(X = k) =$	$\frac{1}{2^5}$	$\frac{5}{2^5}$	$\frac{10}{2^5}$	$\frac{10}{2^5}$	$\frac{5}{2^5}$	$\frac{1}{2^5}$

Espérance. $E(X) = np$.

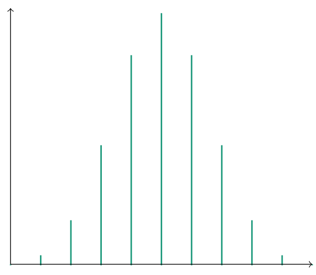
Variance. $V(X) = E(X)q = npq$.

Exemple. Une urne contient N boules dont une proportion p de boules est blanche. On effectue n tirages successifs avec remise et X compte le nombre de boules blanches obtenues dans la série.

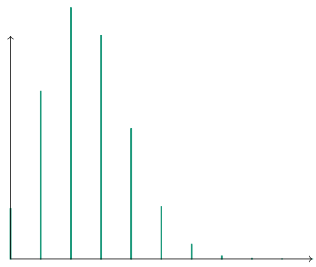
Simulation. Code Python ▼

```python
import random as rd
n,p  = 10,0.3  # loi B(n,p)
# n ép. de Bernoulli indép :
B  = [rd.random()<p for j in
      range(n)]
# On compte les succès :
Nsuc =len([ b for b in B if
      B==1])
```

Peigne.



Loi $\mathscr{B}\left(10, \dfrac{1}{2}\right)$



Loi $\mathscr{B}\left(10, \dfrac{1}{4}\right)$

Remarque. On notera la ressemblance frappante avec une courbe gaussienne dans le cas $p = 1/2$. Cette intuition est précisée lors de l'étude des théorèmes limites.