Lycée Grand Nouméa
ECT 1
A. DA SILVA

TP Python : variables aléatoires

Année 2022

Pour toute la suite, on importera la librairie `numpy` et les sous-librairies `numpy.random` et `matplotlib.pyplot`sous forme d'alias :

```
import numpy.random as rd
import numpy as np
import matplotlib.pyplot as plt
```

L'objectif de ce TP est tout d'abord de simuler des variables aléatoires, c'est-à-dire générer autant de *réalisation* de cette variable que l'on veut (ou presque...). Ensuite on testera les commandes pour construire de jolis diagrammes en barre.

1 Loi uniforme

On souhaite simuler la variable aléatoire qui donne le résultat du lancer d'un dé équilibré à six faces.

- Tester et commenter :

```
rd.seed(5) # Cette commande sert à ..............................
rd.randint(1,7) # Cette commande sert à ......................
```

- Et comparer avec

```
rd.randint(1,7,200)
```

Quelle est la différence ? .....
Que fait la commande `rd.randint(1,7,200)` ? .....
.....

- On va ensuite créer le diagramme en bâton correspondant à 500 lancers :

```
n_lancers=500 # le nombre de lancers
abs=np.arange(0,11,1)
bins=np.arange(0,11,1)-0.5
X=rd.randint(1,7,n_lancers)
plt.hist(X,bins=bins,rwidth=0.6)
plt.xticks(abs)
```

Essayer avec 5000 lancers, puis 20 000. Que constatez-vous ? .....
.....

- Essayez ensuite d'insérer l'option `density=True` et comparer :

```
plt.hist(X,bins=bins,rwidth=0.6,density=True)
```

Que constatez-vous ? .....

2 Loi de Bernouilli

A présent on jette un dé équilibré 20 fois et on s'intéresse à l'évènement « tomber sur un six ». Soit X la variable aléatoire qui compte le nombre de fois où l'on obtient un six au cours de ces 20 lancers.

- La loi suivie par X est .....
- La commande Python qui simule une telle expérience est `rd.binomial(n,p)`. Compléter alors le script suivant pour qu'il simule une réalisation de X.

```
rd.seed(5)
rd.binomial(....,....)
```

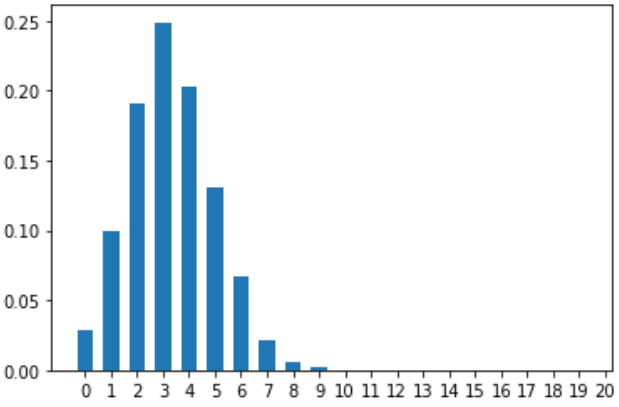
- En vous inspirant de l'exemple précédent, compléter le script suivant pour qu'il génère 1000 fois l'expérience consistant à jeter 200 fois un dé et à compter le nombre de 6.

```
rd.binomial(....,....,.....)
```

- Compléter enfin le script suivant pour qu'il génère les résultats obtenus sous forme d'un diagramme en barre :

```
n_exp=............
abs=np.arange(0,101,1)
bins=np.arange(0,101,1)-0.5
X=rd.binomial(....,....,.....)
plt.hist(X,bins=bins,rwidth=0.6)
plt.xticks(abs)
```

- En insérant l'option `density=True` vous obtenez un diagramme en barre très proche de celui d'une loi binomiale $\mathcal{B}(20, 1/6)$. La forme générale et le « pic »de ce diagramme vous semblent t'ils cohérents ? .....
.....
.....



– Ce que vous devriez obtenir –

3 Diagramme en barre

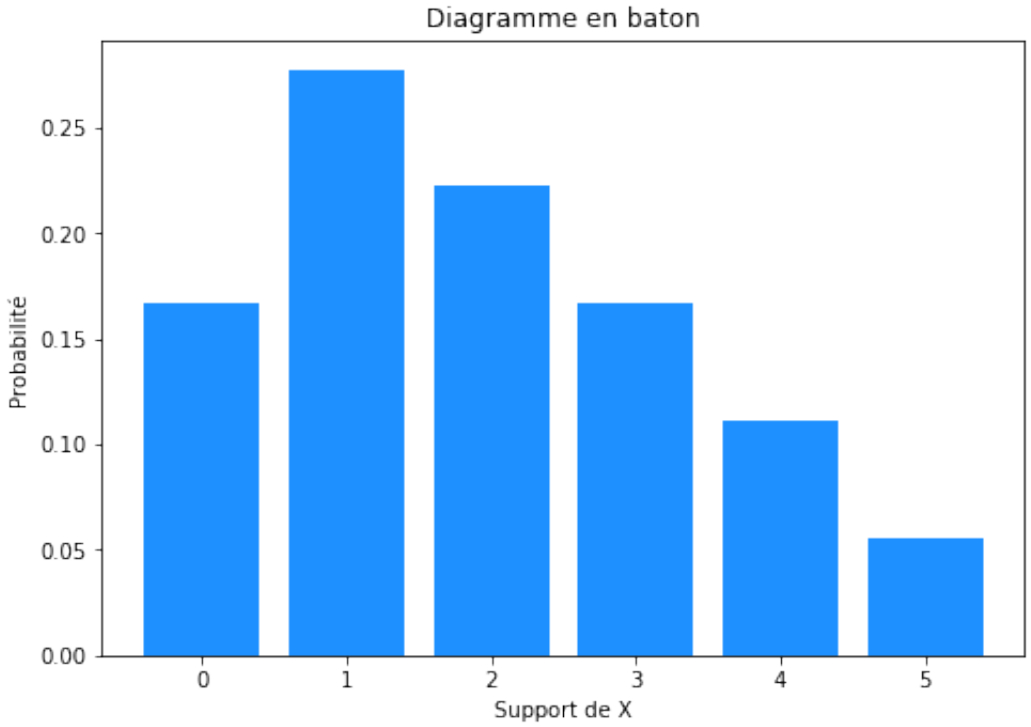
Lorsqu'une loi est connue, on peut la représenter par un diagramme en barre.
Exemple si le tableau de loi est :

x_i	0	1	2	3	4	5
$P(X = x_i)$	$\frac{6}{36}$	$\frac{10}{36}$	$\frac{8}{36}$	$\frac{6}{36}$	$\frac{4}{36}$	$\frac{2}{36}$

Essayez le code suivant :

```
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
abscisse = range(6)
proba_X = [6/36,10/36,8/36,6/36,4/36,2/36]
ax.bar(difference,proba_X,color='dodgerblue')
ax.set_xlabel('Support de X')
ax.set_ylabel('Probabilité')
ax.set_title('Diagramme en baton')
plt.show()
```

Vous devriez obtenir :



4 Comparer deux V.A de même support

Vous pouvez, sur une même figure, faire coïncider plusieurs diagramme en barre. Il suffit d'ajouter autant de commande `ax.bar(.....)` qu'il faut en précisant bien les abscisses et les ordonnées correspondantes. Soit X et Y deux variables aléatoires dont les lois sont donnés par :

x_i	-2	-1	0	1	2
$P(X = x_i)$	0.1	0.15	0.5	0.15	0.1

et

y_i	-2	-1	0	1	2
$P(Y = x_i)$	0.05	0.2	0.55	0.1	0.1

(a) Calculer $E(X)$ et $E(Y)$. Un commentaire ?

(b) Compléter le programme python pour qu'il donne le diagramme en barre des deux variables aléatoires :

```
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
abscisses = np.arange(-2,3) #le tableau de nombres [-2,-1,0,1,2]
proba_X = [.....]
proba_Y = [.....]
ax.bar(.....,.....,color='dodgerblue')
ax.bar(.....,.....,color='r')
ax.set_xlabel('Support de X et de Y')
ax.set_ylabel('Probabilité')
ax.set_title('Diagramme en baton pour X et Y')
plt.show()
```

(c) Le problème est que les diagrammes sont superposés. Pour les décaler, on va décaler le diagramme de Y de 0.25 et réduire la largeur. Remplacer les commandes `ax.bar` par :

```
ax.bar(abscisses,proba_X,color='dodgerblue', width = 0.25)
ax.bar(abscisses+0.25,proba_Y,color='r', width = 0.25)
```

Vous devriez obtenir :

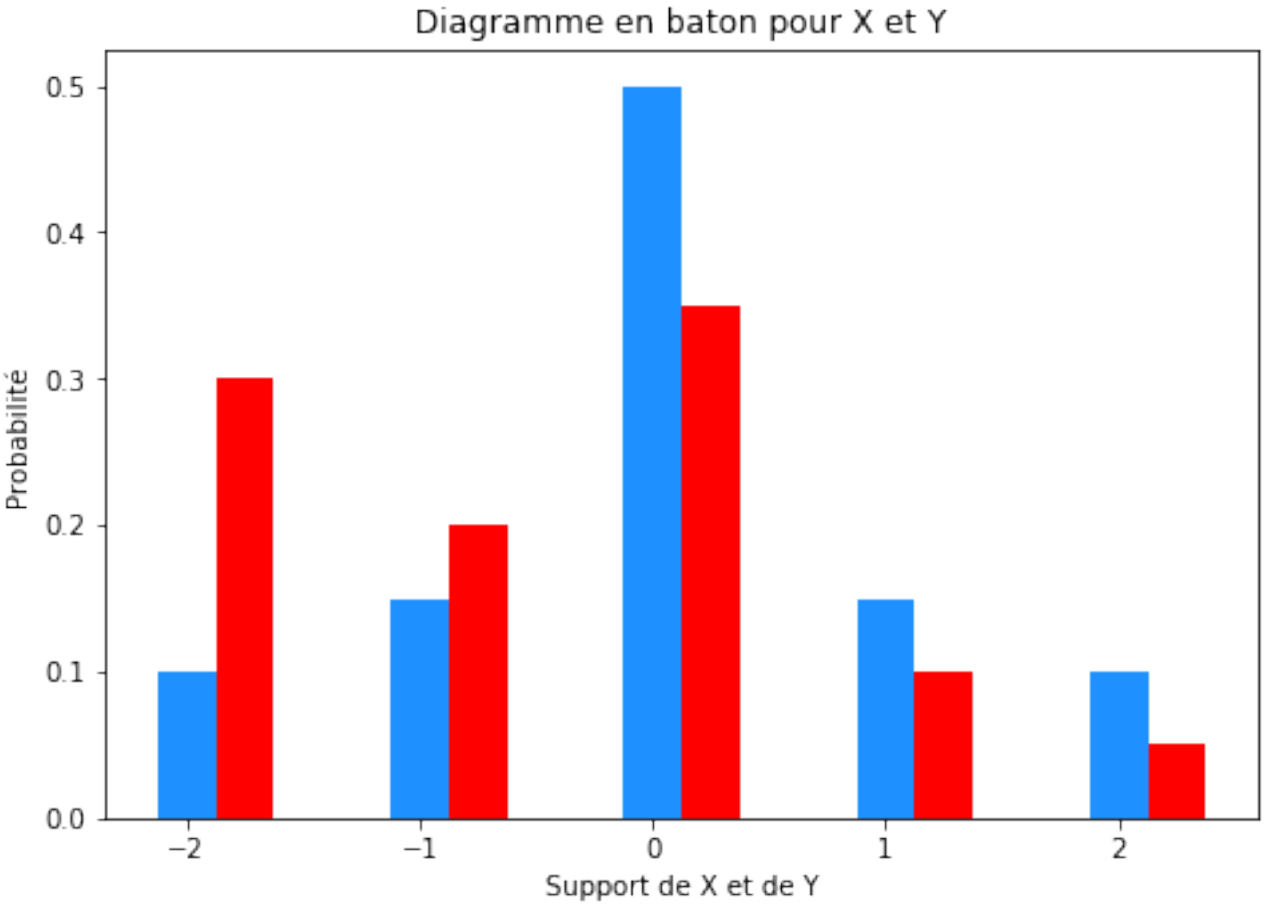


Diagramme en baton pour X et Y
Probabilité
0.0
0.1
0.2
0.3
0.4
0.5
−2
−1
0
1
2
Support de X et de Y