Corrigé Code de partage avec Capytale : 7781-1568933

Exercice 1 - loi uniforme discrète sur $[\![1,6]\!]$

1. Créer un tableau de 10 000 simulations de la loi $\mathscr{U}([\![1,6]\!])$, à l'aide de la librairie `numpy.random`.

   Après avoir importé la bibliothèque import `numpy.random` avec `import numpy.random as rd`, on utilise `rd.randint(1,7,10000)` simule cette expérience qui se résume en « 10 000 lancers d'un dé ».

2. Créer deux tableaux : x contenant les valeurs apparues lors de ces 10 000 simulations et y contenant le nombre d'occurences pour chacune de ces valeurs.

   La commande précédente convient, il faut juste créer une variable qui contient les valeurs : `x=rd.randint(1,7,10000)`.

   Pour les fréquences, on créée un compteur « multiple » et parcourir les 10 000 valeurs pour compter le nombre d'occurences de chaque issue (on utilise une variable auxiliaire ici : `n` et il faut la bibliothèque `numpy` pour utiliser zeros).

```
y=np.zeros(1,6)
for i in range (0,10000):
  n=x[i]
  y[n-1]=y[n-1]+1
```

3. Modifier alors y, afin que y contienne les fréquences d'apparition de ces valeurs.

   Pour passer du nombre d'occurences aux fréquences, il suffit de diviser par 10 000 : `y=y/10000`

4. Faire afficher y à l'aide d'un diagramme en bâtons.

   Il faut d'abord importer `matplotlib.pyplot`, pour utiliser `plt.bar` qui nécessite deux arguments : une liste « d'abscisses » (ici `a` qui contient les entiers de 1 à 6), et une liste « d'ordonnées », on peut ajuster quelques paramètres pour mieux visualiser.

```
a=[k for k in range
   (1,7)]
plt.bar(a,y,width
   =0.7,color='b')
plt.show()
```

5. Quelle est la moyenne de cet échantillon ? Et la variance ?

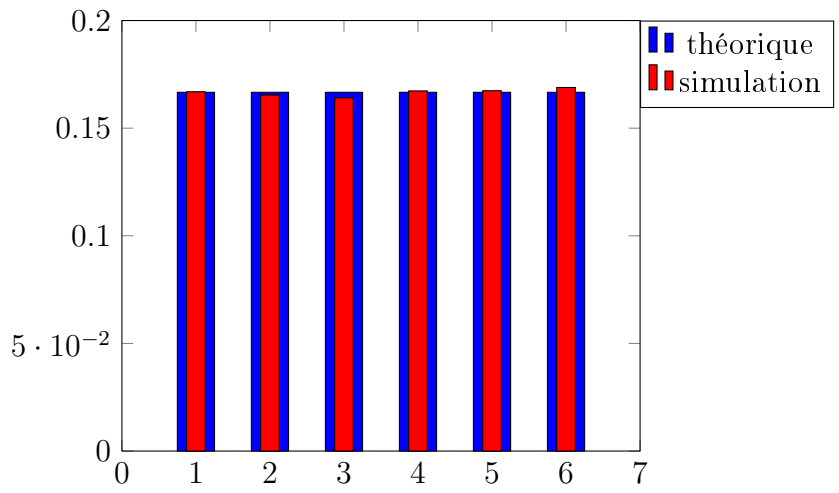
   Pour la moyenne, on utilise `np.mean(x)` et on doit obtenir une valeur proche de la valeur théorique qui est $3,5$.

   Pour la variance, on utilise `np.var(x)` et on doit obtenir une valeur proche de la valeur théorique qui est $\dfrac{35}{12} \simeq 2,92$.

6. Faire afficher le diagramme en bâtons théorique de la $\mathscr{U}([\![1,6]\!])$ (on peut choisir une épaisseur plus petite pour plus de lisibilité). Comparer avec le graphique précédent.

On peut créer le vecteur `z` analogue à `y`, mais contenant les valeurs théoriques, c'est-à-dire $\frac{1}{6}$ pour chaque issue : `z=np.ones(6)/6`. On représente ensuite les deux fréquences par exemple avec `plt.bar(a,z)`, `plt.bar(a,y,width=0.4)` (la deuxième série de valeurs, ici `y`, sera superposée à la première).

On doit obtenir un graphique analogue à celui-ci :



Exercice 2 - loi binomiale $\mathscr{B}(10; 0,3)$

1. Créer un tableau de 10 000 simulations (i.e. un échantillon) de la loi $\mathscr{B}(10; 0,3)$
2. Créer deux tableaux : x contenant les valeurs apparues lors de ces 10 000 simulations et y contenant les fréquences d'apparition pour chacune de ces valeurs.
3. Avec les commandes de `numpy`, repérer la plus petite valeur, la plus grande valeur.
4. Quelle est la moyenne de cet échantillon ? Et la variance ? Comparer avec les valeurs théoriques.
5. Faire alors afficher y avec un diagramme en bâtons.
6. Mémoriser l'allure obtenue en répondant aux questions suivantes : forme de cloche ? symétrique ? centrée autour de quelle valeur ? espérance ? nombre de valeurs prises en pratique ?
7. Enfin, créer le diagramme en bâtons de la loi théorique choisie, pour le comparer au diagramme en bâtons empirique précédent.

On procède comme pour la loi uniforme, sachant que pour la loi binomiale, toutes les valeurs obtenues lors de l'expérience aléatoire doivent être comprises entre 0 et 10, alors qu'avec la loi de Poisson, on ne peut préjuger de l'étendue des valeurs obtenus, d'où l'utilisation du `max` (cf. T.P précédent pour les `import`). On créée une fonction `factorielle` pour calculer les probabilités théoriques.

Pour la loi binomiale :

```
x=rd.binomial(10,0.3,10000)
y=np.zeros(11)

def factorielle (n):
    return np.prod([k for k in
    range(1,n+1)])

for i in range(0, 10000):
    j=x[i]
    y[j]=y[j]+1

y=y/10000 # on passe en
    fréquences
a=[k for k in range(0,11)]
z=[factorielle(10)/factorielle(k
    )/factorielle(10-k)*0.3**k
    *0.7**(10-k) for k in range
    (0,11)]
plt.bar(a,y,width=0.8,color='b')
plt.bar(a,z,width=0.5,color='r')
plt.show()
```

Voici un gaphique que l'on peut obtenir pour l'expérience avec la loi binomiale.