C'est la continuité du TP précédent avec cette fois les lois géométriques et de Poisson.

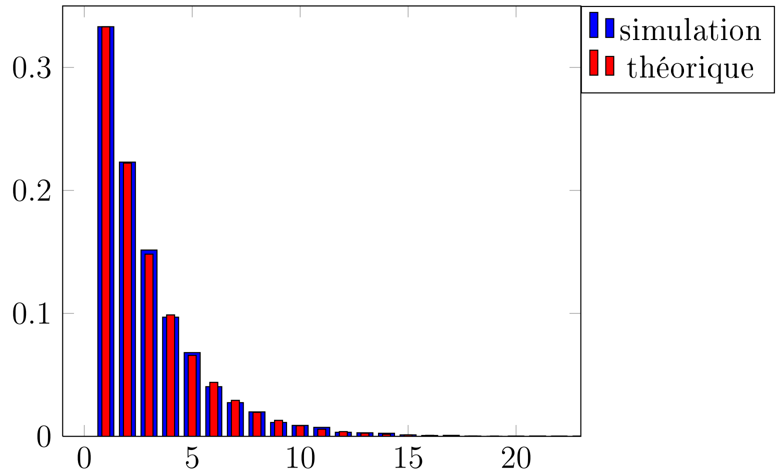
Code de partage avec Capytale : f074-1609150

Exercice 1 - loi géométrique $\mathscr{G}\left(\frac{1}{3}\right)$ et loi de Poisson $\mathscr{P}(5)$

1. Créer un tableau de 10 000 simulations (i.e. un échantillon) de la loi choisie.
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.

Pour la loi géométrique : la démarche est similaire à celle du T.P. précédent, sauf qu'on ne sait pas a priori combien de valeurs seront atteintes (théoriquement le rang du premier succès peut arriver pour n'importe quelle valeur de $\mathbb{N}^*$
On créée donc une variable qui contient la valeur maximale atteinte dans l'expérience : `n=np.max(x)`, puis on utilise un compteur qui calcule le nombre de fois où chaque valeur entre 1 et n est atteinte.
Pour les abscisses, on ajuste pour prendre toutes les valeurs de 1 à n. Pour les valeurs théorique, on ajuste aussi le nombre de valeurs (il y en a n), et on utilise une boucle pour lister les valeurs des probabilités de la loi : $P(X = k) = pq^{k-1}$ pour la loi géométrique.

Voici le gaphique que l'on peut obtenir pour l'expérience avec la loi géométrique.




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

x=rd.geometric(1/3,10000)
n=np.max(x)
y=np.zeros(n)

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

y=y/10000 # on passe en fréquences
a=[k for k in range(1,n+1)]
z=np.zeros(n)
for i in range (0,n):
    z[i]=(2/3)**i*1/3
plt.bar(a,y,width=0.8,color='b')
plt.bar(a,z,width=0.5,color='r')
plt.show()
```

Pour la loi de Poisson :

```
# on définit la fonction factorielle
# pour le calcul des probabilités de
   la loi
def factorielle (n):
    if n==0:
        return 1
    else :
        return n*factorielle(n-1)
x=rd.poisson(5,10000)
n=np.max(x)
y=np.zeros(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,n+1)] # on
   définit la liste d'abscisses (de 0
   à la valeur maximale ici)
z=[5**k*np.exp(-5)/factorielle(k) for
   k in range(0,n+1)]
plt.bar(a,z)
plt.bar(a,y,width=0.5,color='r')
plt.show()
```

Voici un gaphique que l'on peut obtenir pour l'expérience avec la loi de Poisson (attention à partir de 13, Python n'arrive plus à calculer les valeur théoriques (à cause de factorielle).