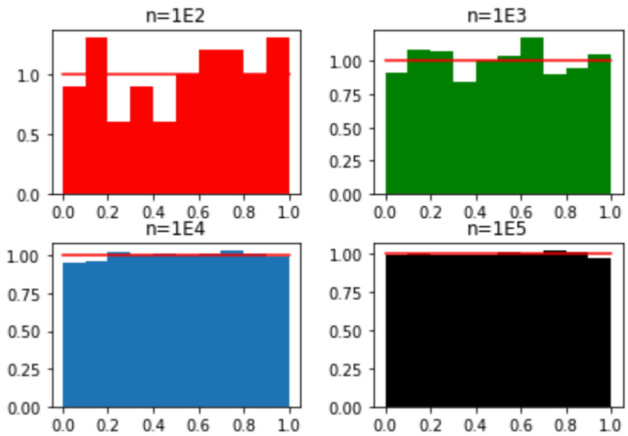
FEUILLE DE TRAVAUX DIRIGÉS N° 1

Pour tous les exercices dans lesquels vous proposerez une méthode de simulation, il est conseillé de faire une vérification par histogramme.

Exercice 1.
Simuler n tirages de loi uniforme sur l'intervalle $]0,1[$, pour n arbitraire. Tracer un histogramme des variables simulées par cette méthode. Comparer, pour différentes valeurs de n, avec un tracé de la densité de la loi uniforme.

Rappel. Après avoir importé `numpy` comme `np` et `matplotlib.pyplot` comme `plt`, utiliser l'instruction `np.random.rand(`n`)` pour la simulation de n v.a. indépendantes de loi uniforme sur $]0,1[$ et l'instruction `plt.hist(x,bins=M,density=True)` pour représenter les valeurs d'un vecteur x sous la forme d'un histogramme avec M classes.



On décompose la fenêtre graphique en quatre sous graphiques et on représente sur chaque sous-graphique un histogramme avec toujours le même nombre de barres (ici 10) mais avec une valeur plus élevée du nombre de tirages n. On prend ainsi successivement n égal à 100, 1000, 10000 et 100000.

```
##Représentation sur une même fenêtre de plusieurs histogrammes

# définition de 4 fenêtres graphiques

fig,ax=plt.subplots(nrows=2,ncols=2)
fig.tight_layout()

# densité pour comparaison

x=np.linspace(0,1,6)
y=np.ones(6)
ticks=('0.0','0.2','0.4','0.6','0.8','1.0')

# cas n=100
ax[0,0].hist(np.random.rand(100), bins = np.arange(0,1.1,.1),density=True,color='r')
ax[0,0].plot(x,y,color='r')
ax[0,0].title.set_text('n=1E2')
ax[0,0].set_xticks(x)
ax[0,0].set_xticklabels(ticks)

# cas n=1000
```

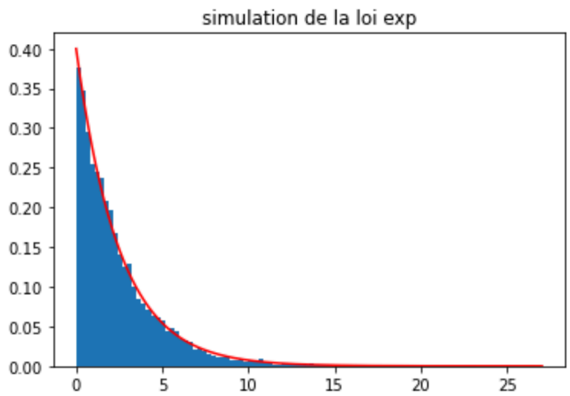
```
ax[0,1].hist(np.random.rand(1000), bins = np.arange(0,1.1,.1),density=True,color='g')
ax[0,1].plot(x,y,color='r')
ax[0,1].title.set_text('n=1E3')
ax[0,1].set_xticks(x)
ax[0,1].set_xticklabels(ticks)

# cas n=10000
ax[1,0].hist(np.random.rand(10000), bins = np.arange(0,1.1,.1),density=True)
ax[1,0].plot(x,y,color='r')
ax[1,0].title.set_text('n=1E4')
ax[1,0].set_xticks(x)
ax[1,0].set_xticklabels(ticks)


# cas n=100000
ax[1,1].hist(np.random.rand(100000), bins = np.arange(0,1.1,.1),density=True,color='k')
ax[1,1].plot(x,y,color='r')
ax[1,1].title.set_text('n=1E5')
ax[1,1].set_xticks(x)
ax[1,1].set_xticklabels(ticks)
plt.show()
```

Exercice 2.

(1) Simuler des variables exponentielles par inversion de la fonction de répartition. On écrira une fonction permettant de réaliser n tirages d'une loi exponentielle de paramètre λ, pour n et λ arbitraires.

(2) Pour un choix de λ, tracer un histogramme des variables simulées par cette méthode. Comparer, pour différentes valeurs de n, avec un tracé de la densité de la loi exponentielle (de paramètre λ)

$$f_\lambda : \begin{array}{ccc} \mathbb{R} & \to & \mathbb{R}^+ \\ x & \mapsto & \lambda e^{-\lambda x}\mathbf{1}_{]0,+\infty[}(x) \end{array}$$

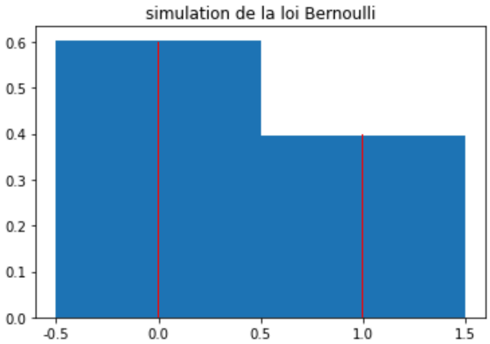



```
## Simulation d'une loi exponentielle

def simuexp(lambd,n):
    r=-np.log(np.random.rand(n))/lambd
    return(r)

#histogramme pour lambda=.4
r=simuexp(.4,10000)
plt.hist(r,100,density=True)
absc=np.arange(0,10,.01)
ordo=.4*np.exp(-.4*absc)
plt.plot(absc,ordo,color='r')
plt.show()
```

Exercice 3.

(1) Simuler des variables de Bernoulli de paramètre p par inversion de la fonction de répartition. On écrira une fonction permettant de réaliser n tirages d'une loi de Bernoulli de paramètre p, pour n et p arbitraires.

(2) Pour un choix de p, représenter les variables simulées sous la forme d'un diagramme en bâtons à l'aide de l'instruction `plt.bar`. Comparer pour différentes valeurs de n.

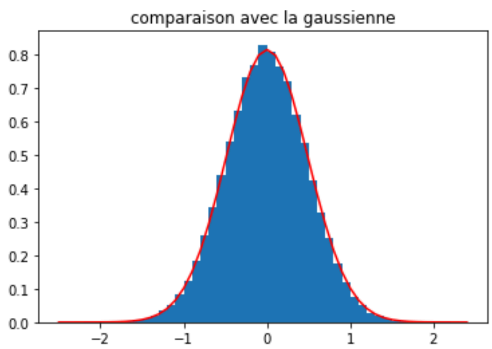



```
##Simulation d'une loi de Bernoulli

def simubernoulli(n,p):
    r=(np.random.rand(n)<p).astype(int)
    return(r)

##representation d'un tirage
r=simubernoulli(1000,.4)
plt.hist(r, bins = (-.5,.5,1.5),density=True)
plt.bar([0,1],[.6,.4],width=.01,color='r')
plt.title('simulation de la loi Bernoulli')
plt.xticks([-.5,0,.5,1,1.5],['-0.5','0.0','0.5','1.0','1.5'])
plt.show()
```

(3) Construire un simulateur d'une loi binomiale de paramètres N et p (attention ici : N est le nombre d'essais dans la loi binomiale). En déduire une fonction permettant de simuler n réalisations indépendantes de loi binomiale de paramètres N et p.

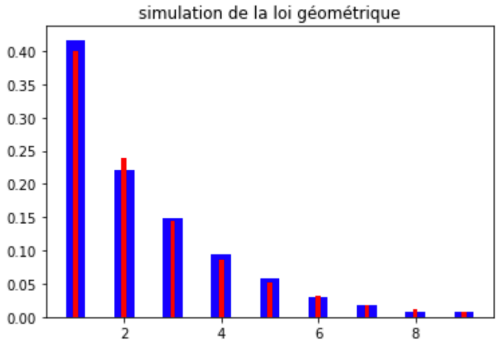
(4) On rappelle que, pour S_N une v.a. de loi binomiale de paramètres N et p, la variable $N^{-1/2}(S_N - Np)$ se comporte, pour N grand, comme une loi gaussienne de moyenne nulle et de variance $p(1-p)$ (théorème de Moivre-Laplace).

Utiliser le simulateur construit dans la question précédente pour mettre en évidence le théorème de Moivre Laplace : on choisira par exemple $p = .4$, $N = 10000$ et on représentera les simulations de $n = 10^6$ réalisations de la loi binomiale sous la forme d'un histogramme et on comparera à la densité gaussienne correspondante.




```
## Simulation de n realisations de loi binomiale B(N,p)

def simubinomiale(n,N,p):
    r=np.repeat(0,n) #vecteur des simulations
    for i in range(n):
        r[i]=np.sum(simubernoulli(N,p))
    return(r)

# comparaison avec la gaussienne

n=100000
N=10000
p=.4
r=(simubinomiale(n,N,p)-p*N)/np.sqrt(N)
plt.hist(r,bins=np.arange(-2.5,2.5,.1),density=True)
absci=np.arange(-2.5,2.5,.1)
ordo=(np.sqrt(2*np.pi*p*(1-p)))**(-1)*np.exp(-(absci**2)/(2*p*(1-p)))
plt.plot(absci,ordo,color='r')
plt.title('comparaison avec la gaussienne')
plt.show()
```

Exercice 4.
Soit X une v.a. de loi géométrique de paramètre p, i.e. $\mathbb{P}(X = k) = p(1-p)^{k-1}$, pour $k \in \mathbb{N}^*$. En utilisant le simulateur de la loi de Bernoulli, construire un simulateur de `n` réalisations indépendantes de X. Comparer avec la loi théorique à l'aide d'un diagramme en bâtons.

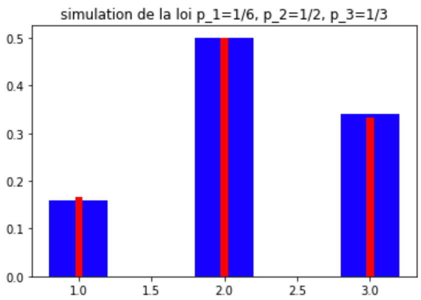



```
## Simulation d'une loi geometrique (1 realisation)
def simugeom(p):
    i=1
    while (simubernoulli(1,p)==0):
        i=i+1
    return(i)

##Construction d'un diagramme en batons
p=.4
n=1000
r=np.zeros(n)
for i in range(n):
    r[i]=simugeom(p)

freq=np.histogram(r, bins =np.arange(.5,10.5,1),density=True)[0]

#diagramme et comparaison avec la loi théorique
plt.bar(np.arange(1,10,1),freq,width=.4,color='b')
plt.bar(np.arange(1,10,1),p*(1-p)**np.arange(0,9,1),width=.1,color='r')
plt.title('simulation de la loi géométrique')
plt.show()
```

Exercice 5.
Construire un simulateur de n v.a. indépendantes $X_1, \cdots, X_n$ de loi

$$\mathbb{P}(X_i = 1) = 1/6, \ \mathbb{P}(X_i = 2) = 1/2, \ \mathbb{P}(X_i = 3) = 1/3.$$

On utilisera l'instruction `cumsum` pour construire la fonction de répartition sous-jacente.




```
## simulation de la loi p_1=1/6, p_2=1/2, p_3=1/3

def simupoids(n):
  F=np.cumsum([1/6,1/2,1/3])
  r=np.random.rand(n)
  simu=(r<F[0]).astype(int)+2*((r<F[1])&(r>=F[0])).astype(int)+3*(r>=F[1]).astype(int)
  return(simu)

r=simupoids(1000)
poids=np.ones(3)
for j in range(3):
    poids[j]=sum(r==(j+1))/n

plt.bar([1,2,3],poids,color='b',width=.4)
plt.bar([1,2,3],[1/6,1/2,1/3],color='r',width=.05)
plt.title('simulation de la loi p_1=1/6, p_2=1/2, p_3=1/3')
plt.show()
```

Exercice 6.

(1) On considère une variable aléatoire X de fonction de répartition F_X et de fonction quantile Q_X. Montrer que si F_X est continue, alors, pour tout $x \in]0,1[$, $F_X(Q_X(x)) = x$. *Attention, cela ne veut pas dire que Q_X est la réciproque de F_X !*

On se fixe $u \in]0,1[$. On veut montrer que $F_X(Q_X(u)) = u$.On sait que F_X est continue de $\mathbb{R}$ dans $[0,1]$. Par ailleurs, on sait que F_X tend vers 0 en $-\infty$ et vers 1 en $+\infty$. Par le théorème des valeurs intermédiaires, on déduit qu'il existe $t \in \mathbb{R}$ tel que $F_X(t) = u$. Par définition de $Q_X(u)$, il vient

$$t \geq Q_X(u).$$

Et, par croissance de F_X, on obtient

$$F_X(t) \geq F_X\big(Q_X(u)\big).$$

Mais, par le lemme de simulation, $F_X(Q_X(u)) \geq u$. On en déduit que

$$u = F_X(t) \geq F_X\big(Q_X(u)\big) \geq u,$$

et donc $F_X\big(Q_X(u)\big) = u$.

(2) On considère la fonction F définie de la manière suivante :

$$F(x) = \begin{cases} \frac{3}{2}x, & \text{si } x \leq 1/3, \\ \frac{1}{2}, & \text{si } 1/3 \leq x \leq 2/3, \\ \frac{3}{2}(x - 2/3) + 1/2, & \text{si } 2/3 \leq x \leq 1. \end{cases}$$

Montrer que F est une fonction de répartition continue mais qu'il existe $x \in \mathbb{R}$ tel que $Q(F(x)) \neq x$ où Q est la fonction quantile associée à F.

On peut vérifier que F est la fonction de répartition de la loi de densité

$$x \mapsto \frac{3}{2}\mathbf{1}_{]0,1/3[}(x) + \frac{3}{2}\mathbf{1}_{]2/3,1[}(x).$$

On calcule Q :

- Si $u \leq 1/2$, $\{x \in \mathbb{R} : F(x) \geq u\} = [\frac{2}{3}u, +\infty[$ et donc $Q(u) = \frac{2}{3}u$.
- Si $u > 1/2$, $\{x \in \mathbb{R} : F(x) \geq u\} = [\frac{2}{3}(u + \frac{1}{2}), +\infty[$ et donc $Q(u) = \frac{2}{3}(u + \frac{1}{2})$.

On choisit maintenant $x = 1/2$. On obtient $F(1/2) = 1/2$ et $Q(F(1/2)) = 1/3$. D'où le contre-exemple recherché ! En fait, cet exemple montre que Q n'est pas continue.