

TP14 : Simulation des lois à densité usuelles, convergence en loi et Théorème central limite

Commencer par importer les bibliothèques suivantes dans chaque fichier **Python** utilisé :

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

On rappelle que l'on a alors accès aux commandes suivantes :

- `rd.random()`
↔ génère un nombre aléatoire dans $[0, 1]$, *i.e.* simule une v.a.r. $X \hookrightarrow \mathcal{U}([0, 1])$
- `rd.random(d)`
↔ génère d nombres aléatoires dans $[0, 1]$
- `rd.random([n,p])`
↔ génère un tableau à n lignes et p colonnes dont chaque case est un nombre aléatoire dans $[0, 1]$
- `rd.exponential(a,d)`
↔ simule d fois une v.a.r. $X \hookrightarrow \mathcal{E}(\frac{1}{a})$ (le paramètre d est optionnel)
- `rd.normal(m,s,d)`
↔ simule d fois une v.a.r. $X \hookrightarrow \mathcal{N}(m, s)$ (le paramètre d est optionnel)

et

- `rd.binomial(n,p,d)`
↔ simule d fois de manière indépendante une v.a.r. $X \hookrightarrow \mathcal{B}(n, p)$ (le paramètre d est optionnel)
↔ en choisissant $n = 1$, simule d fois de manière indépendante une v.a.r. $X \hookrightarrow \mathcal{B}(p)$
- `rd.geometric(p,d)`
↔ simule d de manière indépendante fois une v.a.r. $X \hookrightarrow \mathcal{G}(p)$ (le paramètre d est optionnel)
- `rd.poisson(a,p)`
↔ simule d fois de manière indépendante une v.a.r. $X \hookrightarrow \mathcal{P}(a)$ (le paramètre d est optionnel)
- `rd.randint(a,b,d)`
↔ simule d fois de manière indépendante une v.a.r. $X \hookrightarrow \mathcal{U}(\llbracket a, b-1 \rrbracket)$ (le paramètre d est optionnel)

I. Lien entre histogramme des fréquences et densité

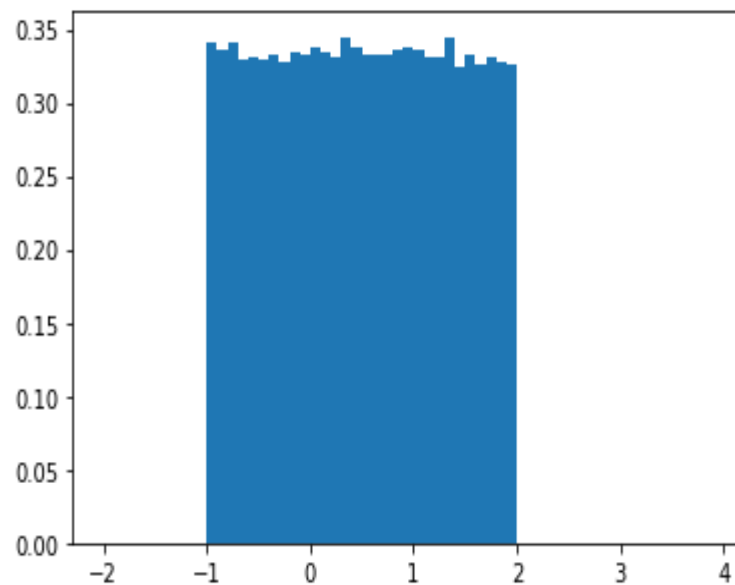
I.1. Simulation de la loi uniforme sur $[a, b]$

- Dessiner le graphe de la densité usuelle d'une v.a.r. $X \hookrightarrow \mathcal{U}([-1, 2])$.



- On représente le tracé obtenu en exécutant le code suivant.

```
1 plt.hist(3*rd.random(100000)-1,np.arange(-2,4,0.1),density=True)
```

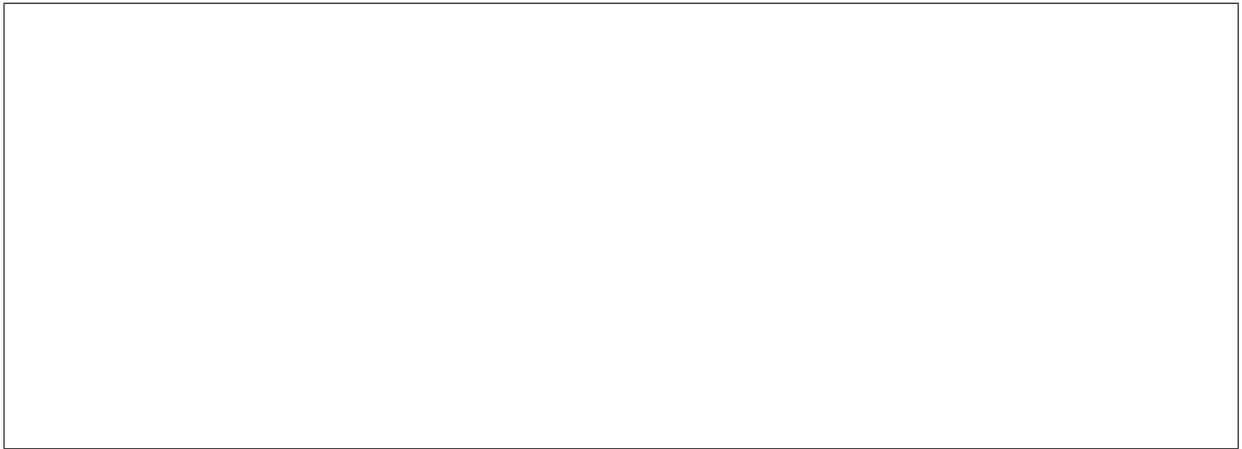


Qu'y a-t-il de remarquable ?

On retrouve la forme de la densité usuelle dessinée ci-dessus (comme si on avait colorié l'aire sous la courbe). Ainsi, on retiendra que l'historgramme des fréquences d'une v.a.r. à densité nous renseigne sur la forme des densités de cette v.a.r. .

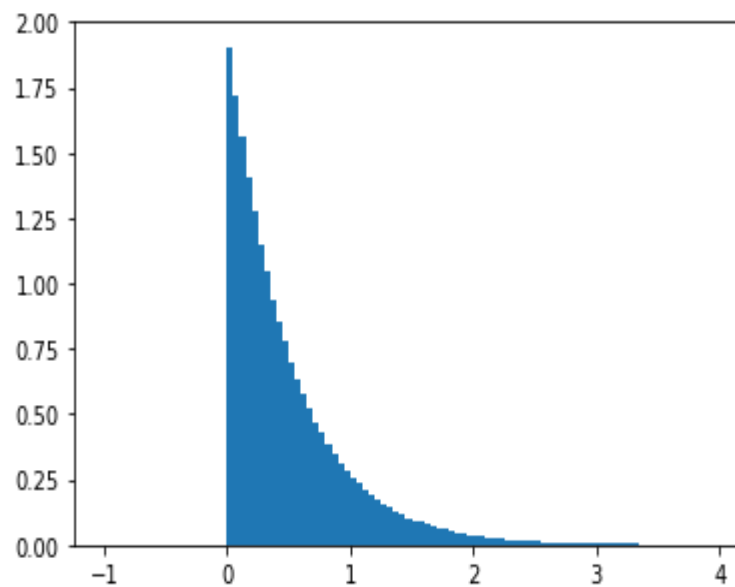
I.2. Simulation de la loi exponentielle

- Dessiner le graphe de la densité usuelle d'une v.a.r. $X \hookrightarrow \mathcal{E}(2)$.



- On représente le tracé obtenu en exécutant le code suivant.

```
1 plt.hist(rd.exponential(1/2,1000000),np.arange(-1,4,0.05),density=True)
```

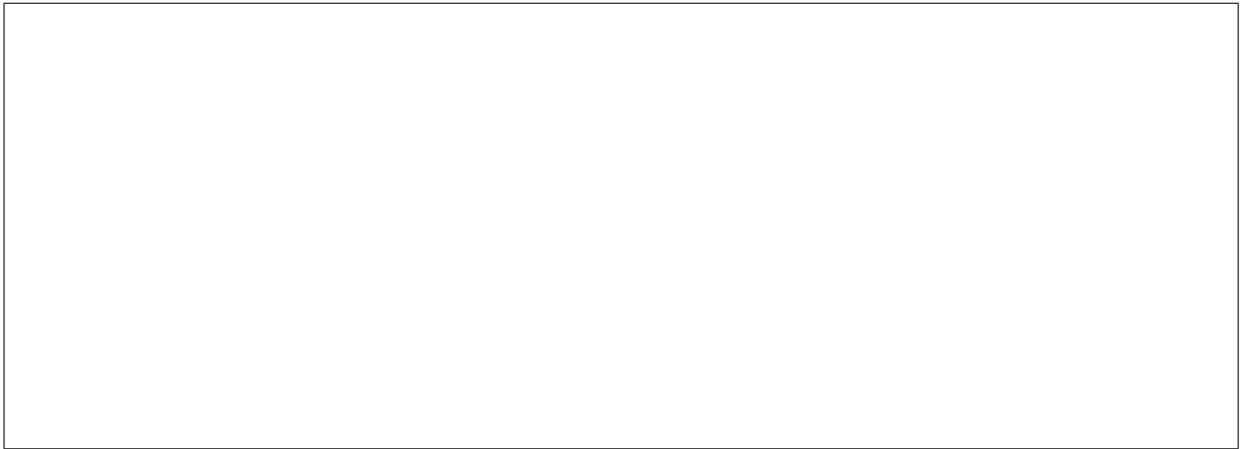


Qu'y a-t-il de remarquable ?

Comme dans la partie précédente, on retrouve la forme de la densité usuelle dessinée ci-dessus (comme si on avait colorié l'aire sous la courbe).

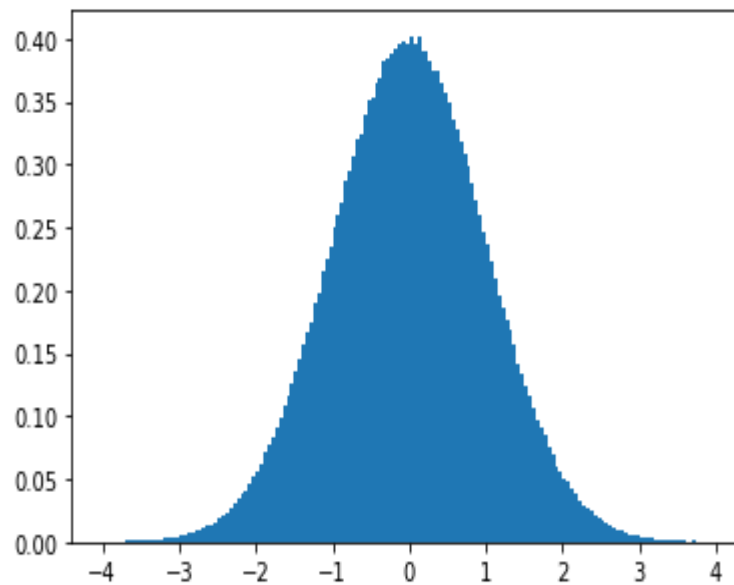
I.3. Simulation de la loi normale centrée réduite

- Dessiner le graphe de la densité usuelle d'une v.a.r. $X \hookrightarrow \mathcal{N}(0, 1)$.



- On représente le tracé obtenu en exécutant le code suivant.

```
1 plt.hist(rd.normal(0,1,1000000),np.arange(-4,4,0.05),density=True)
```



Qu'y a-t-il de remarquable ?

Comme dans la partie précédente, on retrouve la forme de la densité usuelle dessinée ci-dessus (comme si on avait colorié l'aire sous la courbe).

I.4. Visualisation d'une convergence en loi via les histogrammes de fréquences

Soit (U_n) une suite de v.a.r. indépendantes telle que, pour tout $n \in \mathbb{N}^*$, $U_n \hookrightarrow \mathcal{U}([0, 1])$.
 Pour tout $n \in \mathbb{N}^*$, on note $M_n = \max(U_1, \dots, U_n)$ et $X_n = n(1 - M_n)$.

- Compléter la fonction suivante pour qu'elle renvoie une simulation de X_n .

```

1 def SimulX(n) :
2     U = rd.random(n)
3     M = max(U)
4     return n*(1-M)

```

- Compléter la fonction suivante pour qu'elle renvoie un tableau contenant N simulations de X_n .

```

1 def SimulTabX(n,N):
2     T = np.zeros(N)
3     for k in range(N):
4         T[k] = SimulX(n)
5     return T

```

- Expliquer à quoi sert le script suivant.

```

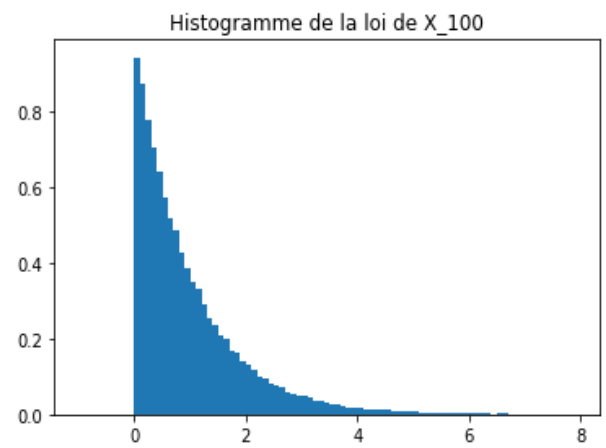
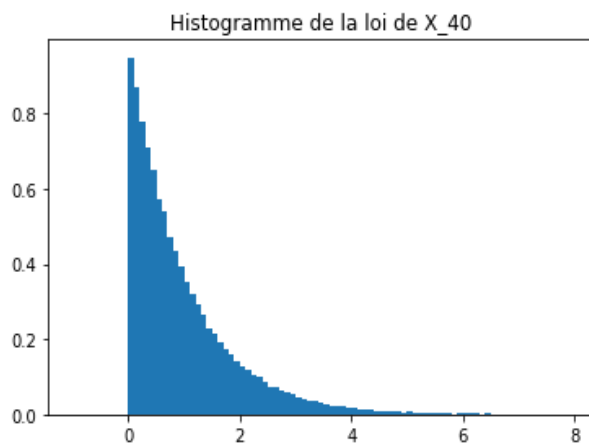
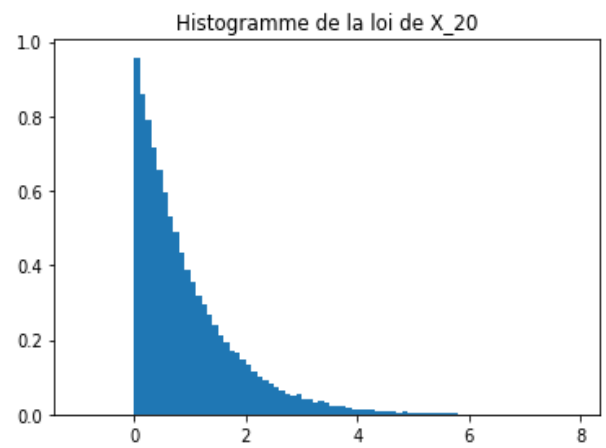
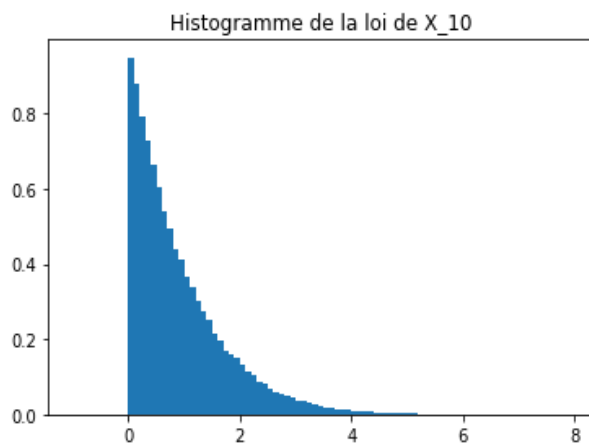
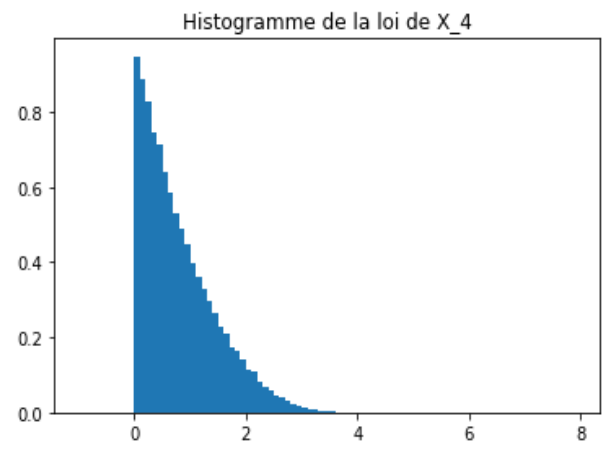
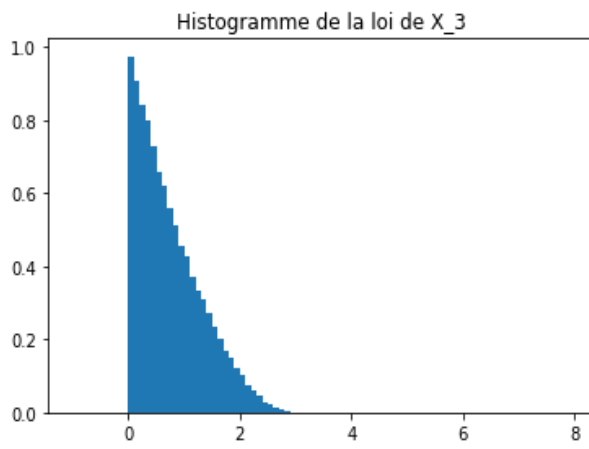
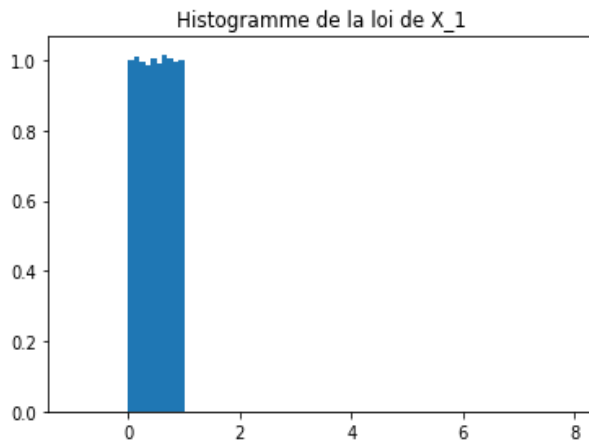
1 for n in [1,2,3,4,10,20,40,100]:
2     plt.hist(SimulTabX(n,100000),np.arange(-1,8,0.1),density=True)
3     plt.title(f'Histogramme de la loi de X_{n}')
4     plt.show()

```

Ce script permet d'afficher les histogrammes de fréquences des lois de X_n pour différentes valeurs de n , plus précisément pour $n \in \{1, 2, 3, 4, 10, 20, 40, 100\}$.

- On reproduit page suivante les résultats obtenus à l'aide du script précédent. Que peut-on conjecturer? La convergence semble-t-elle rapide?

On conjecture que $X_n \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} X$ où $X \hookrightarrow \mathcal{E}(1)$ et la convergence semble rapide (les histogrammes ne changent que très peu à partir de $n = 10$).



II. Théorème central limite

Le cadre du TCL permet de simuler une v.a.r. suivant la loi normale centrée réduite à l'aide d'autres lois usuelles. Dans la suite, on compare la densité de probabilité théorique avec celle obtenue par simulation. Pour ce faire, on procède comme suit :

- × on produit N observations de la simulation de S_n^* ,
- × on trace l'histogramme des fréquences associés.

Il reste à préciser quelle loi doit suivre les variables X_k . On s'intéresse dans la suite aux cas suivants :

- 1) $X_k \hookrightarrow \mathcal{U}([0, 1])$, 2) $X_k \hookrightarrow \mathcal{B}(p)$, 3) $X_k \hookrightarrow \mathcal{P}(a)$.

II.1. Simulation de la loi normale centrée réduite via les « 12 uniformes »

Dans le TCL, la convergence peut être très rapide. La loi de la variable S_n^* est alors une bonne approximation de la loi $\mathcal{N}(0, 1)$ pour des valeurs de n assez petites.

Dans ce paragraphe :

- × on choisit $n = 12$.
- × on considère des variables X_1, \dots, X_{12} indépendantes qui suivent toutes la loi $\mathcal{U}([0, 1])$.
- × on simule la v.a.r. S_{12}^* . Plus précisément, on représente l'histogramme des fréquences issu de N observations de S_{12}^* et on compare ce résultat à la densité de la loi $\mathcal{N}(0, 1)$.

- Que vaut S_{12}^* ? En donner une expression simple à l'aide de $S_{12} = \sum_{k=1}^{12} X_k$.

$$S_{12}^* = \frac{S_{12} - 12 m}{\sigma \sqrt{12}} = \frac{S_{12} - 12 \frac{1}{2}}{\frac{1}{\sqrt{12}} \sqrt{12}} = S_{12} - 6$$

- Par quel appel simule-t-on un échantillon de 12 v.a.r. indépendantes suivant la loi $\mathcal{U}([0, 1])$? En déduire une simulation de la v.a.r. S_{12} .

`rd.random(12)` permet de simuler l'échantillon. Ainsi, `sum(rd.random(12))` simule S_{12} .

- Compléter la fonction suivante pour qu'elle renvoie un tableau contenant N simulations de S_{12}^* .

```

1 def SimulSCRUni(N):
2     T = np.zeros(N)
3     for k in range(N):
4         T[k] = sum(rd.random(12)) - 6
5     return T

```

- Compléter la fonction suivante pour qu'elle renvoie $\varphi(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2}$.

```

1 def phi(t):
2     return (1/np.sqrt(2*np.pi))*np.exp(-t**2/2)

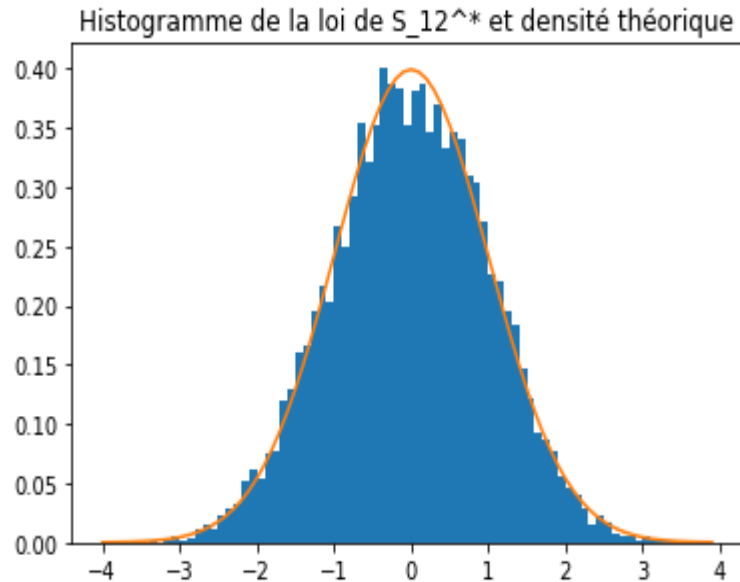
```

- On représente le tracé obtenu en exécutant le code suivant.

```

1 plt.hist(SimulSCRUni(10000),np.arange(-4,4,0.1),density=True)
2 plt.plot(np.arange(-4,4,0.1),phi(np.arange(-4,4,0.1)))
3 plt.title('Histogramme de la loi de S_12^* et densité théorique')
4 plt.show()

```



Commenter.

L'histogramme des fréquences épouse correctement le graphe de la densité théorique φ . C'est remarquable pour une si petite valeur de n ($n = 12$ ici). La convergence du TCL est très rapide dans ce cas.

II.2. Simulation de la loi normale centrée réduite via la loi de Bernoulli

Dans ce paragraphe :

- × on choisit $p = \frac{1}{2}$.
- × on considère des variables X_1, \dots, X_n indépendantes qui suivent toutes la loi $\mathcal{B}(p)$.
- × on simule la v.a.r. S_n^* . Plus précisément, on représente l'histogramme des fréquences issu de N observations de S_n^* et on compare ce résultat à la densité de la loi $\mathcal{N}(0, 1)$.

- Rappeler la loi de $S_n = \sum_{k=1}^n X_k$, son espérance et sa variance.

$$S_n \hookrightarrow \mathcal{B}\left(n, \frac{1}{2}\right), \mathbb{E}(S_n) = \frac{1}{2}n \text{ et } \mathbb{V}(S_n) = \frac{1}{4}n$$

- Que vaut S_n^* ? En donner une expression simple à l'aide de S_n .

$$S_n^* = \frac{S_n - \mathbb{E}(S_n)}{\sigma(S_n)} = \frac{S_n - \frac{1}{2}n}{\frac{1}{2}\sqrt{n}} = \frac{2S_n - n}{\sqrt{n}}$$

- Par quel appel simule-t-on une v.a.r. $Y \hookrightarrow \mathcal{B}\left(n, \frac{1}{2}\right)$?

`rd.binomial(n, 1/2)` permet de simuler $Y \hookrightarrow \mathcal{B}\left(n, \frac{1}{2}\right)$.

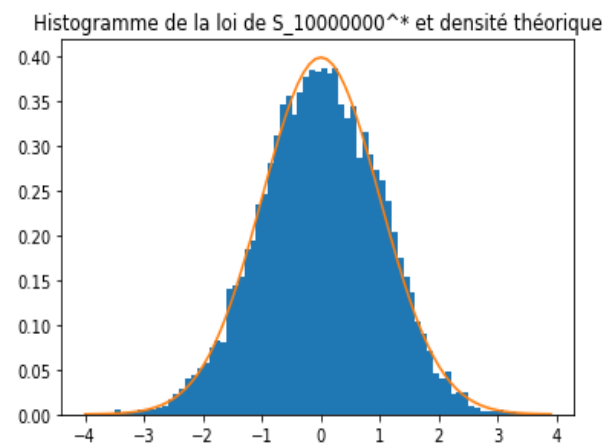
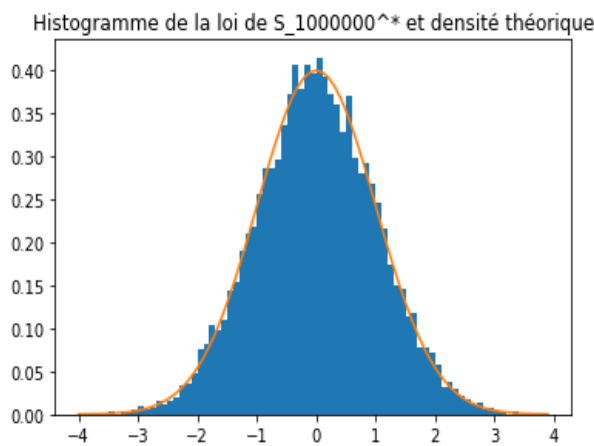
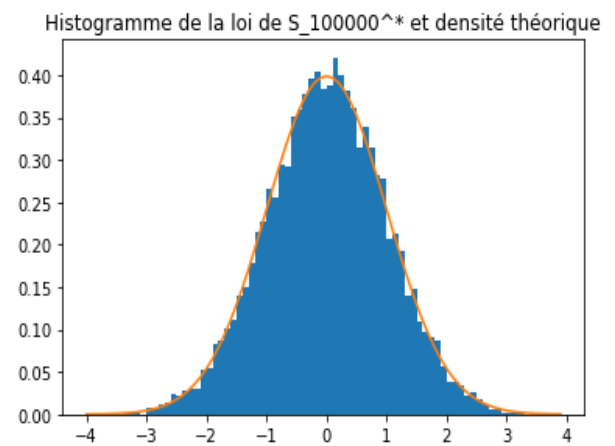
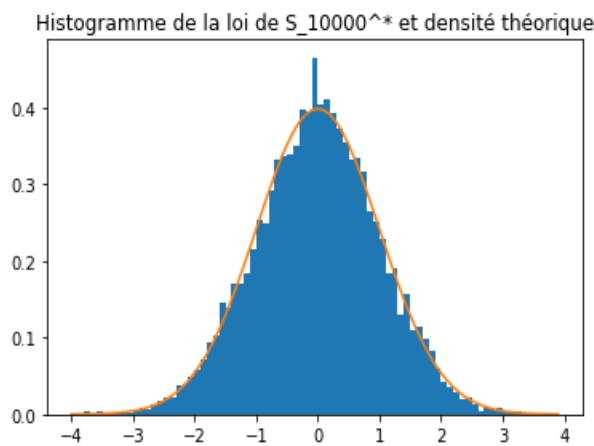
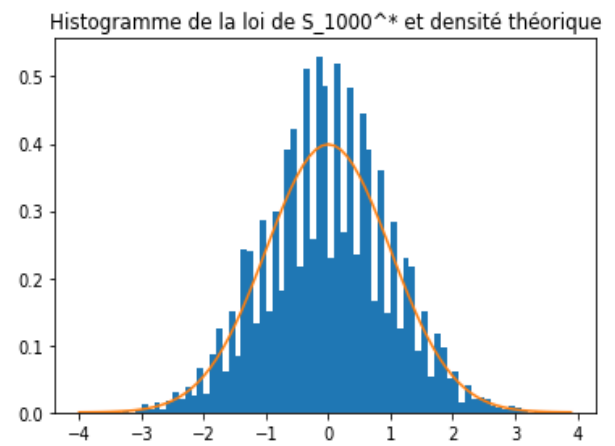
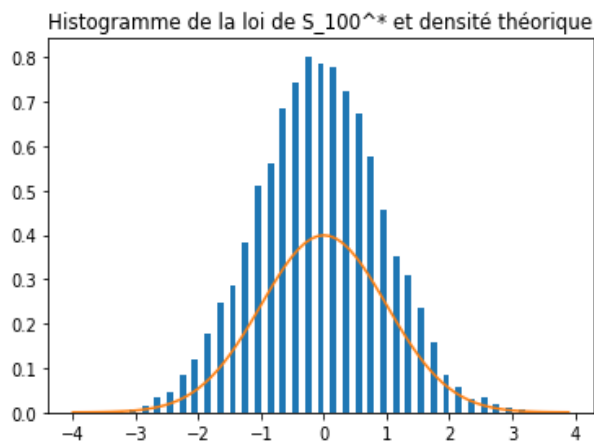
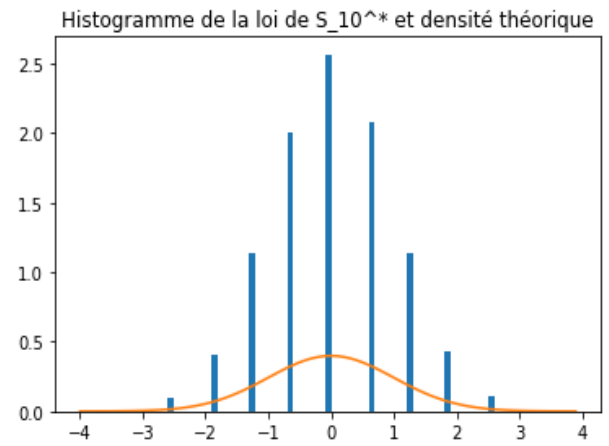
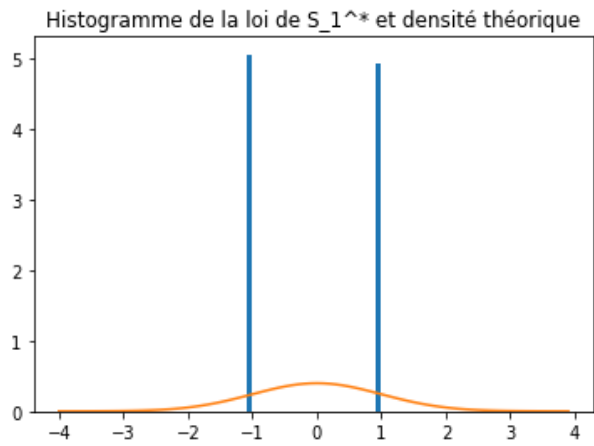
- Compléter la fonction suivante pour qu'elle renvoie un tableau contenant N simulations de S_n^* .

```
1 def SimulSCRBern(n,N):  
2     T = np.zeros(N)  
3     for k in range(N):  
4         T[k] = (2*rd.binomial(n,1/2) - n)/np.sqrt(n)  
5     return T
```

- On reproduit page suivante les résultats obtenus à l'aide du script suivant.
Comparer la vitesse de convergence avec celle de l'exemple précédent.

```
1 for k in range(8):  
2     plt.hist(SimulSCRBern(10**k,10000),np.arange(-4,4,0.1),density=True)  
3     plt.plot(np.arange(-4,4,0.1),phi(np.arange(-4,4,0.1)))  
4     plt.title(f'Histogramme de la loi de S_{10**k} ^* et densité théorique')  
5     plt.show()
```

La convergence est lente comparée à l'exemple précédent. Il faut attendre $n = 10000$ pour avoir une loi ressemblant à la loi normale centrée réduite.



II.3. Simulation de la loi normale centrée réduite via la loi de Poisson

Dans ce paragraphe :

- × on choisit $a = 5$.
- × on considère des variables X_1, \dots, X_n indépendantes qui suivent toutes la loi $\mathcal{P}(a)$.
- × on simule la v.a.r. S_n^* . Plus précisément, on représente l'histogramme des fréquences issu de N observations de S_n^* et on compare ce résultat à la densité de la loi $\mathcal{N}(0, 1)$.
- ▶ Rappeler la loi de S_n , son espérance et sa variance.

$$S_n \hookrightarrow \mathcal{P}(5n), \mathbb{E}(S_n) = 5n \text{ et } \mathbb{V}(S_n) = 5n$$

- ▶ Que vaut S_n^* ? En donner une expression simple à l'aide de S_n .

$$S_n^* = \frac{S_n - \mathbb{E}(S_n)}{\sigma(S_n)} = \frac{S_n - 5n}{\sqrt{5n}}$$

- ▶ Par quel appel simule-t-on une v.a.r. $Y \hookrightarrow \mathcal{P}(5n)$?

`rd.poisson(5n)` permet de simuler $Y \hookrightarrow \mathcal{P}(5n)$.

- ▶ Compléter la fonction suivante pour qu'elle renvoie un tableau contenant N simulations de S_n^* .

```

1 def SimulSCR Pois(n,N):
2     T = np.zeros(N)
3     for k in range(N):
4         T[k] = (rd.poisson(5*n)-5*n)/np.sqrt(5*n)
5     return T

```

- ▶ On reproduit page suivante les résultats obtenus à l'aide du script suivant. Comparer la vitesse de convergence avec celle de l'exemple précédent.

```

1 for k in range(8):
2     plt.hist(SimulSCR Pois(10**k,10000),np.arange(-4,4,0.1),density=True)
3     plt.plot(np.arange(-4,4,0.1),phi(np.arange(-4,4,0.1)))
4     plt.title(f'Histogramme de la loi de S_{10**k} ^* et densité théorique')
5     plt.show()

```

La convergence est un peu plus rapide que dans l'exemple précédent, mais reste beaucoup plus lente que dans le cas des « 12 uniformes ».

