TP 10 : Variables à densité

Exercice 1: *Loi exponentielle*

1. Soit $X \hookrightarrow \mathcal{U}([0,1])$, et soit $\lambda > 0$. Justifier que $T = -\frac{1}{\lambda}\ln(1-X) \hookrightarrow \mathcal{E}(\lambda)$.
2. Ecrire une fonction python qui au paramètre λ renvoie une simulation de la loi $\mathcal{E}(\lambda)$.
3. Représenter l'histogramme empirique de la loi $\mathcal{E}(1)$: on pourra créer une liste S de 10 000 simulations, et utiliser la syntaxe `plt.hist(S,bins=30,density=True)`.
4. Représenter sur un même graphique, la courbe d'une densité de T et l'histogramme. Constater.
   En profiter pour bien re-comprendre ce que représente une densité, et faire également une "analogie" avec le cas discret (lien entre diagramme en bâtons emipirique et théorique).

Exercice 2: *Loi normale*

1. Représenter sur un même graphique plusieurs (au moins 4) densités de loi normale. Visualiser l'influence des deux paramètres de cette loi sur la densité. Bien comprendre le lien avec l'espérance et la variance.
2. La syntaxe pour simuler une variable de loi normale centrée réduite est : `rd.gauss(0,1)`.
   Comment en déduire (à l'aide du cours de mathématiques) une simulation d'une loi normale de paramètre μ et σ^2 ?

Remarque : la syntaxe `rd.gauss(`μ`,`σ`)` permet de simuler directement la loi normale $\mathcal{N}(\mu,\sigma^2)$ (attention, le second paramètre est σ et non σ^2), mais n'est pas dans votre mémo.

Exercice 3: *Méthode de Monte-Carlo*

Soit $X \hookrightarrow \mathcal{U}([0,1])$ et $Y = 4\sqrt{1-X^2}$.

1. Montrer que Y admet une espérance et que $E(Y) = \pi$.
   On pourra utiliser le changement de variable x=sin t.
2. En déduire une méthode probabiliste pour calculer une valeur approchée de π.

Exercice 4:

Le but de l'exercice est d'illustrer le théorème central limit (qui sera étudié dans le dernier chapitre de mathématiques de l'année). L'idée du théorème est la suivante :
Soit n un entier naturel non-nul et $(X_1, X_2, ..X_n)$ n variables aléatoires indépendantes, de même loi, admettant une espérance notée $\boxed{m}$ et une variance notée $\boxed{s^2}$. Alors si n est suffisamment grand,

$$\boxed{\text{la loi de la variable } M_n = \frac{X_1 + ... + X_n}{n} \text{ est très proche de la loi normale } \mathcal{N}(m, \tfrac{s^2}{n})}.$$

Ce résultat précise la LFGN : en effet, dans le TP 9, on a pu constater que M_n "s'approchait" de m.
Ce nouveau résultat permet de comprendre que les fluctuations de M autour de m se comportent comme la loi normale $\mathcal{N}(0, \frac{s^2}{n})$.

1. Quelle est l'espérance et la variance théorique de la variable M_n ? En déduire un moyen mnémotechnique pour retenir les paramètres de la loi normale qui approche la loi de M_n d'après le théorème ci-dessus.
2. Cas où $(X_1, X_2, ..X_n)$ sont n variables aléatoires de même loi $\mathcal{G}(0.2)$.
   (a) Donner la valeur de m et de s^2 dans ce cas.
   (b) Créer une fonction python qui simule une variable de loi $\mathcal{G}(0.2)$.
   (c) Créer une fonction python qui simule M_n, lorsque $n = 30$.
   (d) Tracer un histogramme des valeurs obtenues de M_n après 10 000 simulations.
   On pourra utiliser la syntaxe `plt.hist()` de l'exercice 1 (avec le même nombre de classes).
   (e) D'après le théorème central limit, quels sont les paramètres μ et σ de la loi normale qui approche la loi de M_n ?
   Ajouter alors le tracé de la fonction densité de cette loi normale : on pourra faire ce tracé sur l'intervalle $[\mu - 3\sigma, \mu + 3\sigma]$. Constater
   (f) Ne pas hésiter à changer la valeur de n pour voir son influence : $n = 50$, $n = 100$...
3. Cas où $(X_1, X_2, ..X_n)$ sont n variables aléatoires de même loi uniforme sur $[-2, 2]$
   (a) A l'aide de la syntaxe `rd.random()`, simuler une variable de loi uniforme sur $[-2, 2]$.
   (b) Reprendre ensuite les questions ci-dessus, et les adapter dans ce nouveau cas.

Exercice 5: *Fonction de répartition empirique*

1. Ecrire une fonction Python d'arguments une liste L et un réel x qui renvoie la proportion d'éléments de L inférieurs ou égaux à x
2. En utilisant cette fonction :
   (a) Représenter la fonction de répartition empirique d'une loi uniforme $\mathcal{U}([0,1])$. La comparer à la fonction de répartition théorique. *On pourra se placer sur un intervalle du type* $[-1, 2]$.
   (b) Représenter la fonction de répartition empirique d'une loi normale centrée réduite.
   (c) Représenter la fonction de répartition empirique d'une loi binomiale puis d'une loi géométrique (fixer des paramètres au choix). En profiter pour réviser les propriétés des fonctions de répartition de variables discrètes !
3. Soit $(X_k)_{k\in\mathbb{N}^*}$ des variables aléatoires indépendantes de même loi uniforme $\mathcal{U}([0,1])$. On pose pour tout $n \in \mathbb{N}^*$, $Y_n = n(1 - max(X_1, ..., X_n))$.
   (a) Pour réviser : écrire une fonction Python max_pos, qui prend en entrée une liste L et qui renvoie la valeur maximale de L ainsi que la position de cette valeur maximale dans la liste.
   (b) Réaliser une simulation numérique de Y_{30}.
   (c) Tracer la fonction de répartition empirique de Y_{30}.
   (d) Rajouter sur le graphique précédent la fonction de répartition empirique d'une loi exponentielle de paramètre 1. *Au vu du (c), ne pas hésiter à restreindre l'intervalle d'étude.*
   Que constatez-vous ? Prendre alors $n = 100$. Constater.
   (e) Pour information : lorsque pour tout $x \in \mathbb{R}$ (ou presque), $F_{X_n}(x) \underset{n\to+\infty}{\longrightarrow} F_X(x)$ on dira que la suite de variables (X_n) converge en loi vers la variable X. Qu'avez-vous validé informatiquement dans la question précédente ?

Exercice 6: *Approximation de la loi binomiale* $\mathcal{B}(n, p)$ *par une loi normale*

1. Soit $X \hookrightarrow \mathcal{B}(n, p)$. La loi binomiale étant une répétition de n épreuves de Bernoulli indépendantes, on émet l'hypothèse selon laquelle la loi de X pourrait être approchée par une loi normale (même esprit que le théorème central limite précédent, même si X n'est pas une moyenne M_n).
   Sous cette hypothèse, quels pourraient être les paramètres de cette loi normale ?
2. Ecrire une fonction python qui simule une variable aléatoire de loi $\mathcal{B}(30, 0.6)$.
3. Créer une fonction qui crée une liste de 10 000 simulations de loi $\mathcal{B}(30, 0.6)$, puis créer l'histogramme correspondant.
   On pourra utiliser la syntaxe `plt.hist()` avec `bins=20`.
4. A l'aide de la syntaxe `rd.gauss(`μ`,`σ`)`, créer un histogramme empirique de la loi normale avec les paramètres trouvés à la question 1. Penser à l'option `width=` dans `plt.hist` pour que ce deuxième histogramme ne cache pas le premier, et prendre 20 classes également (`bins=20`).
5. L'hypothèse énoncée lors de la question 1., est-elle plausible dans notre cas ?