TP9B - ESTIMATIONS

Dans tout le TP, on importe les modules suivants :

```python
import numpy as np
import numpy.random as rd
import scipy.special as sp
import matplotlib.pyplot as plt
```

1 Estimations ponctuelles

Exercice 1 ⋆⋆

Une variable X suit une loi $\mathcal{B}(p)$ où p est un paramètre inconnu. Nous allons étudier le comportement de quelques estimateurs dans ce contexte.
On commence par prendre au hasard une valeur de p :

```python
p = rd.random()
```

1. On simule 10 000 tirages selon la loi correspondante :

```python
X = rd.binomial(1,p,10000)
```

   Représenter la loi estimée à partir des simulations. Que semble valoir p ? Est-ce correct ?
2. On commence avec l'estimateur $\overline{X_k}$ de la moyenne empirique des tirages (X_i). Compléter la fonction :

```python
def estimations_p(p,n):
  estimations = np.zeros(n)
  for i in range(n):
    X = rd.binomial(1,p,100)
    estimations[i] = ...
  return estimations
```

   En utilisant $n = 10\ 000$, représenter la loi de $\overline{X_k}$.
3. Écrire une fonction simulant cette fois des estimations avec :

$$S_n = \frac{1}{n}\sum_{k=1}^{n}(X_i - \overline{X_n})^2.$$

   Tracer la loi. Quelle est l'espérance estimée ? Quelle est la maximum de probabilité observé ? Comparer avec la valeur exacte de la variance attendue.
4. Reprendre la question précédente avec :

$$\tilde{S_n} = \frac{1}{n-1}\sum_{k=1}^{n}(X_i - \overline{X_n})^2.$$

2 Estimations par intervalle de confiance

Exercice 2 - Calcul de t_α ⋆

Afin de déterminer des intervalles de confiance, nous aurons besoin d'inverser la fonction Φ (fonctiond de répartition de la loi normale centrée réduite), notamment pour trouver le réel t_α tel que $\Phi(t_\alpha) = 1 - \frac{\alpha}{2}$.
Pour cela, nous allons utiliser la fonction `sp.ndtri` qui prend en paramètre un réel p et renvoie x tel que $\Phi(x) = p$ (la fonction `sp.ndtr` fait l'inverse : elle prend x et renvoie p).

1. Vérifier rapidement que $\Phi(0) = \frac{1}{2}$ puis que `sp.ndtri` renvoie des valeurs négatives pour des antécédents inférieurs à $\frac{1}{2}$ et positives pour des antécédents supérieurs à $\frac{1}{2}$.
2. Déterminer numériquement $t_{0,05}$ et $t_{0,01}$.

Exercice 3 - Un intervalle de confiance ⋆⋆⋆

On peut montrer que si $X_1, \ldots, X_n$ suit une loi normale $\mathcal{N}(\theta, 1)$ alors $\left[\overline{X_n} - \frac{t_\alpha}{\sqrt{n}}, \overline{X_n} + \frac{t_\alpha}{\sqrt{n}}\right]$ est un intervalle de confiance de θ au niveau de confiance $1 - \alpha$.
Choisissons θ au hasard avec le code suivant :

```
theta = rd.exponential(1/100)
```

On prendra $\alpha = 0,05$ et on crée un tableau contenant 10 000 échantillons de $X_1, \ldots, X_{10}$:

```
X = rd.normal(theta,1,10000)
```

1. Créer deux tableaux A et B tels que, pour tout i, l'intervalle de confiance associée au $i^{\text{ème}}$ échantillon soit `[A[i],B[i]]`.
2. Parmi les 10 000 échantillons, quelle proportion vérifie vraiment $\theta \in [A[i], B[i]]$?
3. Écrire une fonction `def etendue(n,alpha):` qui prend en paramètre un entier n et un réel $\alpha \in]0, 1[$ et renvoie l'étendue de l'intervalle de confiance de θ correspondant.

   Essayer ce programme avec $\alpha = 0,05$ en faisant varier n et constater l'effet que cela a sur l'étendue de l'intervalle.

   De même, à n fixé, étudier l'étendue de l'intervalle pour $\alpha = 0,05$, $\alpha = 0,01$, $\alpha = 0,001$ et $\alpha = 0,0001$.