
TD 3

RANDOM – BRUITAGE

Pour l'ensemble des TD suivants, on préférera l'utilisation de programmes plutôt que l'éditeur python lui-même. `numpy.random` est module permettant de traiter des variables aléatoires et faire des opérations stochastiques.

Exercice 1. Les bases avec random

L'objectif de cet exercice est de visualiser avec python les fonctions de répartition (courbe) et densité de probabilité (histogrammes) des lois statistiques courantes.

1. Importer le module `numpy.random` ainsi que le module `matplotlib.pyplot` permettant de créer des graphiques. Créer une liste `x` de $N = 100$ entiers entre 0 et 9 espacés régulièrement.
2. Créer un vecteur `y_uni` contenant N nombres tirés aléatoirement suivant une loi uniforme entre 0 et 10: `uniform(a,b, N)`
3. Tracer `y_uni` en fonction de `x`. On représentera aussi les N valeurs triées (`list.sort()`).
4. Tracer également l'histogramme représentant le vecteur avec 20 bins: `plt.hist(list, 20)`
5. Reprendre les 3 questions précédentes avec un vecteur `y_norm` contenant N nombre tirés aléatoirement suivant une loi gaussienne de moyenne 4 et d'écart-type 1.5 `normal(moy, std, N)`. Qu'observez-vous ?
6. Faites les mêmes opérations pour une loi triangulaire de paramètres 0 (min), 4 (center) et 9 (max): `triangular(a,b,c, N)` et une loi exponentielle de paramètre $\lambda = 4$: `exponential(lambd)`
7. Donner les moyennes théoriques de chacune des lois.
8. Tracer les 4 courbes sur le même graphe. Observer les valeurs obtenues pour chacune des lois en $x = 4$. Commenter.

Exercice 2. Bruitage

1. Récupérer le signal `td3.wav` puis l'ouvrir avec la fonction `fs, stereo = wavfile.read('td3.wav')`. Vous pourrez l'écouter mais pas en TD! Donner sa fréquence d'échantillonnage.
2. Le convertir de stereo vers mono en ne prenant qu'une seule piste puis normaliser son amplitude en divisant par l'amplitude maximum du signal original (`np.max(stereo)`).
ATTENTION: le vecteur `stereo` obtenu ne contient que des entiers. La division peut alors entraîner quelques soucis, à vous de régler le problème.
3. Créer un vecteur de temps correspondant à la durée du signal et sa fréquence d'échantillonnage et tracer le signal en fonction du temps à partir de 1 sec sur 1000 échantillons. Qu'observez-vous ?
4. Créer un vecteur `bruit` de valeurs prises aléatoirement suivant une loi uniforme entre -0.05 et 0.05. Le nombre de valeurs de ce vecteur doit être égal au nombre d'échantillons du signal audio.

5. Bruiter le signal `mono` en ajoutant le signal de bruit. Tracer ce nouveau signal. Vous pouvez également créer un fichier wav et l'écouter mais je vous le déconseille fortement! Vous ne maîtrisez pas l'amplitude du signal bruité généré et le volume pourrait détruire vos oreilles!. Qu'en concluez-vous ?
6. Modifier le vecteur de bruit en prenant des valeurs qui suivent une loi normale de moyenne 0 et d'écart-type 0.01. Tracer le nouveau signal.

Exercice 3. Théorème central limite

1. Tirer 10000 valeurs suivant une loi normale centrée réduite et tracer l'histogramme des valeurs obtenues en découpant l'intervalle $[-2;2]$ en 100 sous-intervalles. Quelle courbe obtient-t-on ? Représenter la courbe théorique de la loi normale sur le graphique obtenu précédemment.
2. Tirer de même 100 valeurs suivant une loi binomiale et tracer l'histogramme correspondant. Augmenter progressivement le nombre de valeurs tirées. Que remarquez-vous ?