TP13/14 : Illustration du Théorème Central Limite

Pré-requis : je vous invite à consulter les chapitres de cours correspondants sur ma page (support informatique). Pour ce TP, on pourra en particulier se reporter à la section « Tracés d'histogrammes » du CH 8.

▶ Dans votre dossier `Info_2a`, créez le dossier `TP_13`.

I. Théorème Central Limite

Le but de ce TP est d'illustrer le **Théorème Central Limite**[1] (TCL). Ce théorème énonce la **convergence en loi** de suites de variables aléatoires. Commençons par rappeler la définition de convergence en loi.

Définition

Soit $(X_n)_{n\in\mathbb{N}^*}$ une suite de v.a.r. définies sur $(\Omega, \mathscr{A}, \mathbb{P})$.

Soit X une v.a.r. définie sur $(\Omega, \mathscr{A}, \mathbb{P})$.

Soient F_{X_n} et F_X les fonctions de répartition de associées à ces v.a.r.

- On dit que la suite $(X_n)_{n\in\mathbb{N}^*}$ **converge en loi** vers X si :

$$\lim_{n\to+\infty} F_{X_n}(x) = F_X(x)$$

en tout point $x \in \mathbb{R}$ où F_X est continue.

- On note alors : $X_n \xrightarrow{\mathcal{L}} X$.

Théorème 1. *Théorème Central Limite*

Soit $(X_n)_{n\in\mathbb{N}^*}$ une suite de v.a.r. indépendantes, de même loi, d'espérance m et d'écart-type σ.

Notons $S_n = \sum_{k=1}^{n} X_k$ *et* $\overline{X_n} = \dfrac{S_n}{n}$.

Notons enfin $S_n{}^* = \dfrac{S_n - nm}{\sigma\sqrt{n}} = \dfrac{\overline{X_n} - m}{\frac{\sigma}{\sqrt{n}}}$ *la v.a.r. centrée réduite associée à S_n.*

- *Alors $(S_n{}^*)$ converge en loi vers une v.a.r. de loi normale $\mathcal{N}(0,1)$.*
- *En particulier, pour tout $a \in \overline{\mathbb{R}}$ et tout $b \in \overline{\mathbb{R}}$ tels que $a \leqslant b$, on a :*

$$\lim_{n\to+\infty} \mathbb{P}(a \leqslant S_n{}^* \leqslant b) = \int_a^b \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}\, dt$$

☞ Ainsi, pout n suffisamment grand, la loi de $S_n{}^*$ approche la loi $\mathcal{N}(0,1)$.
Nous allons illustrer cette propriété pour des suites de v.a.r. (X_n) dont tous les éléments suivent une loi uniforme / de Bernoulli / de Poisson.

[1] On parle aussi du Théorème de la Limite Centrée (TLC).

Dans la suite, on considère les suites de v.a.r. (X_n) et (S_n) comme définies dans l'énoncé.

▶ Pour $n \in \mathbb{N}^*$, déterminer l'espérance et la variance de S_n.

- $\mathbb{E}(S_n) = \mathbb{E}(\sum_{k=1}^{n} X_k) = \sum_{k=1}^{n} \mathbb{E}(X_k) = \sum_{k=1}^{n} m = n\ m$ *(par linéarité de l'espérance)*
- $\mathbb{V}(S_n) = \mathbb{V}(\sum_{k=1}^{n} X_k) = \sum_{k=1}^{n} \mathbb{V}(X_k) = \sum_{k=1}^{n} \sigma^2 = n\ \sigma^2$ *(par indépendance)*

▶ Déterminer l'espérance et la variance de ${S_n}^*$.

- $\mathbb{E}({S_n}^*) = \mathbb{E}\left(\dfrac{S_n - nm}{\sigma\sqrt{n}}\right) = \dfrac{\mathbb{E}(S_n - nm)}{\sigma\sqrt{n}} = \dfrac{\mathbb{E}(S_n) - nm}{\sigma\sqrt{n}} = \dfrac{nm - nm}{\sigma\sqrt{n}} = 0$ *(centrée)*
- $\mathbb{V}({S_n}^*) = \mathbb{V}\left(\dfrac{S_n - nm}{\sigma\sqrt{n}}\right) = \dfrac{\mathbb{V}(S_n - nm)}{\sigma^2\ n} = \dfrac{\mathbb{V}(S_n)}{\sigma^2\ n} = \dfrac{n\ \sigma^2}{\sigma^2\ n} = 1$ *(réduite)*

▶ Déterminer $\overline{X_n}^*$ la v.a.r. centrée réduite associée à X_n.

$\mathbb{E}(\overline{X_n}) = \mathbb{E}\left(\dfrac{S_n}{n}\right) = \dfrac{\mathbb{E}(S_n)}{n} = \dfrac{n\ m}{n} = m$ et $\mathbb{V}(\overline{S_n}) = \mathbb{V}\left(\dfrac{S_n}{n}\right) = \dfrac{1}{n^2}\ \mathbb{V}\,(S_n) = \dfrac{n\ \sigma^2}{n^2} = \dfrac{\sigma^2}{n}$

Ainsi, $\overline{X_n}^* = \dfrac{\overline{X_n} - m}{\sqrt{\frac{\sigma^2}{n}}} = \dfrac{\overline{X_n} - m}{\frac{\sigma}{\sqrt{n}}} = {S_n}^*$

II. La loi normale en Scilab

▶ Coder dans un onglet **SciNotes** la densité d'une loi normale centrée réduite.
On nommera cette fonction `densiteNormaleCR`.

```
function y = densiteNormaleCR(x)
    y = 1/(sqrt(2*%pi)) * exp(-x^2/2)
endfunction
```

▶ Rappeler l'expression de la fonction de répartition Φ de la loi normale centrée réduite.

Φ n'admet pas d'expression « simple ». On revient donc à la définition.

$$\forall x \in \mathbb{R},\ \Phi(x) = \int_{-\infty}^{x} \varphi(t)\ dt = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}}\ \mathrm{e}^{-\frac{t^2}{2}}\ dt$$

▶ Quelle méthode (non aléatoire) peut-on proposer pour calculer une version approchée de cette fonction de répartition ?

Il s'agit d'obtenir un calcul approché d'intégrales.
On peut le faire via la méthode des rectangles / trapèzes.

En **Scilab**, ce calcul approché est implémenté par la fonction `cdfnor` (**c**umulative **d**istribution **f**unction **nor**mal distribution). L'appel général est le suivant :

```
[P,Q] = cdfnor("PQ", X, Mean, Std)
```

Détaillons les différents éléments de cet appel.

× `X` : borne supérieure d'intégration,

× `Mean` : moyenne choisie (la loi n'est pas forcément centrée),

× `Std` : écart-type choisi (la loi n'est pas forcément réduite),

× `P` : résultat de l'appel. Autrement dit, c'est le calcul approché de $\int_{-\infty}^{\texttt{X}} \varphi(t)\ dt$,

× `Q` : second résultat de l'appel. Cette variable contient le calcul approché de $\int_{\texttt{X}}^{+\infty} \varphi(t)\ dt$.

▶ Donner une relation entre `P` et `Q`.

> On a : `Q` $= 1 -$ `P`.

▶ Coder dans un nouvel onglet **SciNotes** la fonction de répartition de loi normale centrée réduite. On nommera cette fonction `repartitionNormaleCR`.

```
1 function y = repartitionNormaleCR(x)
2     [y,u] = cdfnor("PQ", x, 0, 1)
3 endfunction
```

Remarque

- L'option `"PQ"` sert à signifier que l'on cherche à déterminer les paramètres `P` et `Q`. Pour ce faire, tous les autres paramètres doivent être fournis lors de l'appel.
- En fait, `cdfnor` permet de réaliser le calcul de n'importe lequel des paramètres de la distribution normale si tous les autres paramètres sont fournis à la fonction. Plus précisément, on peut écrire :
  - `X = cdfnor("X", Mean, Std, P, Q)`,
  - `Mean = cdfnor("Mean", Std, P, Q, X)`,
  - `Std = cdfnor("Std", P, Q, X, Mean)`.

III. Simulation de loi normale à l'aide de lois usuelles

Le cadre du TCL permet de simuler une v.a.r. suivant la loi normale centrée réduite à l'aide de 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)* $X_k \hookrightarrow \mathcal{P}(\lambda)$.

III.1. Simulation de la loi normale 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.

III.1.a) Rappels sur la loi uniforme

▶ Si $(a,b) \in \mathbb{R}^2$ avec $a < b$, que signifie $X \hookrightarrow \mathcal{U}([a,b])$?

a) $X(\Omega) = [a,b]$

b) X admet pour densité la fonction :

$$f : \left| \begin{array}{ccc} \mathbb{R} & \to & \mathbb{R} \\ x & \mapsto & \left\{ \begin{array}{ll} 0 & \text{si } x < a \\ \frac{1}{b-a} & \text{si } x \in [a,b] \\ 0 & \text{si } x > b \end{array} \right. \end{array} \right.$$

▶ Rappeler la valeur de $\mathbb{E}(X)$ et $\mathbb{V}(X)$ si $X \hookrightarrow \mathcal{U}([a,b])$.

- La v.a.r. X admet une espérance ssi l'intégrale impropre $\displaystyle\int_{-\infty}^{+\infty} t f(t)\, dt$ est absolument convergente. Cela revient à démontrer de la convergence pour les calculs de moment. Remarquons alors :
$$\int_{-\infty}^{+\infty} t f(t)\, dt = \int_a^b t\, f(t)\, dt$$
car la fonction f est nulle en dehors de $[a,b]$.
La fonction $t \mapsto t\, f(t)$ est continue par morceaux sur **le segment** $[a,b]$. On en déduit que l'intégrale impropre considérée est bien définie. Ainsi, la v.a.r. X admet une espérance.
De plus :
$$\begin{aligned} \mathbb{E}(X) &= \int_a^b t\, f(t)\, dt = \int_a^b \frac{t}{b-a}\, dt \\ &= \frac{1}{b-a}\int_a^b t\, dt = \frac{1}{b-a}\left[\frac{t^2}{2}\right]_a^b \\ &= \frac{1}{2}\,\frac{b^2-a^2}{b-a} = \frac{1}{2}\,\frac{\cancel{(b-a)}(b+a)}{\cancel{b-a}} = \frac{a+b}{2} \end{aligned}$$
- On procède de même pour le calcul de $\mathbb{E}(X^2)$.
$$\begin{aligned} \mathbb{E}(X^2) &= \int_a^b \frac{t^2}{b-a}\, dt = \frac{1}{b-a}\int_a^b t^2\, dt = \frac{1}{b-a}\left[\frac{t^3}{3}\right]_a^b \\ &= \frac{1}{3}\,\frac{b^3-a^3}{b-a} = \frac{1}{3}\,\frac{\cancel{(b-a)}(a^2+ab+b^2)}{\cancel{b-a}} = \frac{a^2+ab+b^2}{3} \end{aligned}$$
- Ainsi, $\mathbb{V}(X) = \mathbb{E}(X^2) - (\mathbb{E}(X))^2 = \dfrac{(b-a)^2}{12}$.

▶ En déduire la valeur de $m = \mathbb{E}(X)$ et $\sigma^2 = \mathbb{V}(X)$ dans le cas où $X \hookrightarrow \mathcal{U}([0,1[)$.

On a alors : $\quad m = \mathbb{E}(X_k) = \dfrac{1}{2} \quad$ et $\quad \sigma^2 = \mathbb{V}(X_k) = \dfrac{1}{12} \quad$ soit $\quad \sigma = \dfrac{1}{\sqrt{12}}$.

III.1.b) Simulation de la densité de probabilité

Dans ce paragraphe :

× on considère des variables X_k indépendantes telles que $X_k \hookrightarrow \mathcal{U}([0,1[)$.

× on choisit $n = 12$. Autrement dit, $S_n = S_{12}$.

× on simule la v.a.r. ${S_{12}}^*$. Plus précisément, on représente l'histogramme des effectifs issu de N observations de ${S_{12}}^*$ et on compare ce résultat à la densité de la loi $\mathcal{N}(0,1)$.

► Que vaut alors ${S_{12}}^*$? En donner une expression simple à l'aide de S_{12}.

On a alors : ${S_{12}}^* = \dfrac{S_{12} - 12\ m}{\sigma\ \sqrt{12}} = \dfrac{S_{12} - 12\ \frac{1}{2}}{\frac{1}{\sqrt{12}}\ \sqrt{12}} = S_{12} - 6$

► Par quel appel simule-t-on, à l'aide de la fonction `grand`, `1` échantillon de `12` v.a.r. indépendantes suivant la loi $\mathcal{U}([0,1[)$? En déduire la simulation de la v.a.r. S_{12}.

`G = grand(1, 12, "def")` et `S = sum(G)`

► Par quel appel simule-t-on, à l'aide de la fonction `grand`, `N = 10000` échantillons de `12` v.a.r. indépendantes suivant la loi $\mathcal{U}([0,1[)$? En déduire la simulation de `N` échantillons de S_{12}.

Il y a deux manières de procéder.

- Soit on reprend ce qui précède et on remplit alors un tableau de taille `N` à l'aide d'une boucle.
- Soit on se sert des fonctionnalités **Scilab** et de `grand` en particulier.
  Plus précisément :
  `N = 10000, G = grand(N, 12, "def")` et `S = sum(G, 'c')`

► Recopier et compléter le programme suivant.
On l'enregistrera sous le nom `comparaison_dnormale_12_unif.sce`.

```
// Valeur des paramètres
N = 10000
nbC = 100

// Simulations de N observations de S12
G = grand(N, 12, "def")
S = sum(G, 'c')

// Simulations de N observations de S12*
Scr = S - 6

// Tracé de la densité théorique
plot(linspace(-5,5),densiteNormaleCR)
// Tracé de la densité observée
histplot(nbC, Scr)

legend(["Simu - 12 uniformes","Densité N(0,1)"], "in_upper_right")
xtitle(["Théorème Central Limite :";
" diagramme des fréquences des 12 uniformes";
" comparaison avec le tracé de la densité de N(0,1)"])
```

III.1.c) Simulation de la fonction de répartition

On se propose maintenant de comparer la fonction de répartition théorique avec sa version obtenue par simulation. Pour ce faire, on procède comme suit :

× on récupère le tableau des effectifs des N observations précédentes,

× on trace le diagramme en bâtons des effectifs **cumulés** associé.

× compare ce diagramme avec la fonction de répartition de la loi $\mathcal{N}(0,1)$.

On pourra faire appel à la fonction `dsearch` qui permet de déterminer la fréquence des valeurs contenues dans un vecteur. L'appel général est le suivant :

```
[indClasse, effectif] = dsearch(Obs, classe)
```

Détaillons les différents éléments de cet appel.

× `Obs` : une série d'observations,

× `classe` : une liste de réels rangés dans l'ordre strictement croissant définissant les classes,
Par exemple, `classe = [0, 3, 7, 9]` *permet de définir trois classes :*

1) *la* $1^{\text{ère}}$ *classe est l'intervalle* `[0, 3]`,

2) *la* $2^{ème}$ *classe est l'intervalle* `]3,7]`,

3) *la* $3^{ème}$ *classe est l'intervalle* `]7,9]`.

× `indClasse` : indique, pour chaque observation, la classe à laquelle elle appartient,

× `effectif` : indique, pour chaque classe, le nombre d'observations qu'elle contient.

▶ Commenter le résultat de l'appel :

```
[indClasse, effectif] = dsearch([8.4, 7, 0, 1, 9, 8, %pi, 2, sqrt(2)], [0, 3, 7, 9])
```

> On obtient :
>
> - `indClasse = [3, 2, 1, 1, 3, 3, 2, 1, 1]` ce qui signifie que `8.4` est dans la classe `3`, `7` est dans la classe `2`, `0` est dans la classe `1`, `1` est dans la classe `1`, ..., `sqrt(2)` est dans la classe `1`.
> - `effectif = [4, 2, 3]` ce qui signifie qu'il y a 4 observations dans la $1^{\text{ère}}$ classe (il s'agit de `0`, `1`, `2` et `sqrt(2)`), 2 observations dans la $2^{\text{ème}}$ classe (il s'agit de `7` et `%pi`) et 3 observations dans la $3^{\text{ème}}$ classe (il s'agit de `8.4`, `9`, `8`).

▶ Par quel appel obtient-on le tableau `effCumule` des effectifs cumulés à l'aide du tableau `effectif` des effectifs ?

> Il suffit de faire appel à la fonction `cumsum`.
> Plus précisément : `effCumule = cumsum(effectif)`.

▶ Comment obtenir la fréquence des observations dans chaque classe ?

> - Pour obtenir le tableau des fréquences, il suffit de diviser le tableau des effectifs par le nombre d'observations au total. Ce qui correspond à l'appel :
>
> ```
> effectif / length(Obs)
> ```
>
> - Évidemment, si l'on souhaite obtenir le tableau des fréquences cumulées, il suffit de diviser le tableau des effectifs cumulés par le nombre d'observations au total.
>
> ```
> effCumule / length(Obs)
> ```

▶ La fonction `dsearch` est l'analogue, dans le cas continu, d'une fonction dont on s'est servi dans le cas discret. Rappeler le nom de cette fonction et décrire brièvement son utilisation.

- La fonction `dsearch` est l'analogue de la fonction `tabul`.
- Étant donné un tableau d'observations `Obs`, la fonction `tabul` renvoie un tableau contenant deux colonnes.

  1) La première colonne liste les différents réels présents dans le tableau `Obs`.
  Ces réels apparaissent dans l'ordre croissant si on ajoute l'option `"i"` *(increasing)*.

  2) La deuxième colonne liste le nombre d'appartions de ces réels dans `Obs`.

  Par exemple, `tabul([5, 5, 1, 5, 3, 1, 2.2, 2.2, 5], "i")` renvoie :

1.	2.
2.2	2.
3.	1.
5.	4.

▶ Recopier et compléter le programme suivant.
On l'enregistrera sous le nom `comparaison_rnormale_12_unif.sce`.

```
1  // Valeur des paramètres
2  N = 10000
3  nbC = 100
4
5  // Simulations de N observations de S12
6  G = grand(N, 12, "def")
7  S = sum(G, 'c')
8
9  // Simulations de N observations de S12*
10 Scr = S - 6
11
12 // Simulation de la fonction de répartition
13 classe = linspace(-10, 10, nbC+1)
14 [indice, effectif] = dsearch(Scr, classe)
15 effCumule = cumsum(effectif)
16
17 // Tracé de la fonction de répartition observée
18 bar(classe(1:nbC), effCumule/N, width=1)
19 // Tracé de la fonction répartition théorique
20 plot(classe, répartitionNormaleCR)
21
22 legend(["Simu - 12 uniformes","Fonction répartition N(0,1)"],
23 "in_upper_left")
24 xtitle(["Théorème Central Limite :";
25 " diagramme des fréquences cumulées des 12 uniformes";
26 " comparaison avec le tracé de la fct répartition de N(0,1)"])
```