TP8 : Simulation de loi via la fonction `grand`

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_8`.

I. Utilisation du générateur `grand`

Dans ce TP, nous délaissons la fonction `rand` pour utiliser la fonction `grand` qui présente l'avantage de pouvoir simuler très simplement l'observation d'un échantillon de `N` v.a.r. indépendantes.

I.1. Simulation d'une v.a.r. suivant une loi géométrique

I.1.a) Utilisation de la fonction `grand`

- Évaluer plusieurs fois la commande `grand(1,1,"geom",0.4)`.
  Qu'obtient-on ? À quoi sert la commande `grand` munie du paramètre `"geom"` ?

> - On obtient la séquence de valeurs suivantes : 4, 1, 5, 4, 1, 7, 5, 1 ...
> - La commande `grand(1,1,"geom",0.4)` permet de simuler une v.a.r. X suivant la loi géométrique de paramètre 0.4. On rappelle que si $X \hookrightarrow \mathcal{G}(p)$, on a :
>   × $X(\Omega) = \mathbb{N}^*$
>   × $\forall k \in \mathbb{N}^*, \ \mathbb{P}(X = k) = (1-p)^{k-1} \times p$

- Évaluer maintenant la commande `grand(1,10,"geom",0.4)`. Qu'obtient-on ?

> - On obtient la matrice ligne de taille 1×10 suivante : $[2, 1, 1, 2, 1, 1, 2, 10, 7, 12]$.
> - La commande `grand(1,10,"geom",0.4)` permet de simuler un échantillon de 10 v.a.r. indépendantes X_i suivant toutes la loi géométrique de paramètre 0.4.

- Évaluer maintenant la commande `grand(2,4,"geom",0.4)`. Qu'obtient-on ?

> - On obtient la matrice de taille 2×4 suivante : $\begin{pmatrix} 7 & 1 & 7 & 7 \\ 7 & 3 & 8 & 1 \end{pmatrix}$.
> - La commande `grand(2,4,"geom",0.4)` permet de simuler 2 échantillons de 4 v.a.r. indépendantes X_i suivant toutes la loi géométrique de paramètre 0.4.

- Généraliser ce résultat.

> La commande `grand(m,N,"geom",p)` permet de simuler `m` échantillons de `N` v.a.r. indépendantes X_i suivant toutes la loi géométrique de paramètre `p`.

► Écrire un programmme qui :

× demande à l'utilisateur d'entrer la valeur d'un paramètre `p`,

× demande à l'utilisateur d'entrer la valeur d'un paramètre `N`,

× réalise la simulation d'un échantillon de `N` v.a.r. indépendantes de loi géométrique $\mathcal{G}(\mathtt{p})$,
(on stocke le résultat obtenu dans une variable `Obs`*)*

× trace le diagramme des effectifs de cette simulation.

On pourra utiliser les fonctions `tabul` (pour calculer les effectifs) et la fonction `bar` (pour représenter le digramme en bâtons correspondant).

```
p = input("Entrez la valeur du paramètre p : ")
N = input("Entrez la valeur de N : ")

Obs = grand(1, N, "geom", p)
v = tabul(Obs)

clf()
bar(v(:,1), v(:,2))
```

(on enregistre ce programme sous le nom `distrib_geom.sce`*)*

► Ajouter `width=0.4,'red'` dans l'appel à la fonction `bar`. À quoi servent ces arguments optionnels ?

- Le paramètre `width` permet de préciser la largeur des bâtons.
- Le paramètre `'red'` permet de préciser la couleur (rouge) des bâtons.

► Comment modifier le programme précédent afin d'obtenir le diagramme des fréquences ?

Le vecteur `v(:,2)` contient les effectifs observés. En divisant ces effectifs par le nombre total d'observations (`N`), on obtient un vecteur de fréquences.

On remplace l'appel à `bar` par : `bar(v(:,1), v(:,2)/N, width=0.4, 'red')`.

I.1.b) Comparaison avec la distribution théorique

On propose d'ajouter les lignes suivantes au programme précédent.

```
x = 1:10
y = (1-p) ^ (x-1) * p

bar(x, y, width=0.4)
```

► Que contient la variable `y` ? Que cela représente-t-il pour une v.a.r. X telle que $X \hookrightarrow \mathcal{G}(p)$?

La variable `y` contient ici `[p, (1-p)*p, (1-p) ^ 2*p, (1-p) ^ 3*p, ..., (1-p) ^ 10*p]`.

Autrement dit : $[\mathbb{P}(X = 1), \mathbb{P}(X = 2), \mathbb{P}(X = 3), \ldots, \mathbb{P}(X = 10)]$.

▶ Le choix fait pour la variable `x` vous semble-t-il pertinent ?
Comment la modifier de sorte à représenter autant de bâtons que dans le diagramme précédent ?

- Le support d'une v.a.r. X telle que $X \hookrightarrow \mathcal{G}(p)$ est : $X(\Omega) = \mathbb{N}^*$.
  Comme on ne peut représenter toutes les valeurs de $\mathbb{P}(X = k)$ pour k dans $\mathbb{N}^*$, on fait ici le choix de s'arrêter à $k = 10$. L'idée est que `(1-p)^k` devient rapidement faible.
  Évidemment, ceci dépend du choix de `p`.
- Pour dessiner le même nombre de bâtons que dans le premier appel à `bar` il faut trouver l'abscisse la plus élevée des bâtons précédents. On peut procéder comme suit :

```
m = max(v(:, 1)) // abscisse maximale
x = 1:m
```

▶ Comment s'affiche les deux diagrammes obtenus ? Comment améliorer cet affichage ?

- Les diagrammes se superposent : les bâtons des deux diagrammes sont dessinés aux mêmes abscisses.
- Pour améliorer la lisibilité, on peut décaler le diagramme de la distribution approchée en remplaçant l'appel à `bar` par :

```
bar(v(:,1)+.5, v(:,2)/N, width=0.4, 'red')
```

▶ On pourra enfin ajouter un titre et une légende à ce diagramme à l'aide des commandes suivantes, ajoutées en fin de programme.

```
legend(["Diagramme distribution approchée","Diagramme distribution théorique"],
"in_upper_right")

xtitle("Comparaison de la distribution théorique et de la distribution approchée
pour une var suivant une loi géométrique")
```

I.2. Simulation d'une v.a.r. suivant une loi de Poisson

I.2.a) Utilisation de la fonction `grand`

▶ Si `m`, `N` et `lam` sont donnés, à quoi sert l'appel `grand(m,N,"poi",lam)` ?

La commande `grand(m,N,"poi",lam)` permet de simuler `m` échantillons de `N` v.a.r. indépendantes X_i suivant toutes la loi de Poisson de paramètre `lam`.

I.2.b) Comparaison avec la distribution théorique

▶ Soit X une v.a.r. . Que signifie que $X \hookrightarrow \mathcal{P}(\lambda)$? Détailler.

- $X(\Omega) = \mathbb{N}$
- $\forall k \in \mathbb{N},\ \mathbb{P}(X = k) = \mathrm{e}^{-\lambda} \times \dfrac{\lambda^k}{k!}$

- Dans un nouvel onglet **SciNotes**, copier le programme `distrib_geom.sce` et l'adapter au cas de la loi de Poisson. Pour la distribution théorique, on pourra se servir de la fonction `factorial`.

```
// Valeur des paramètres
lam = input("Entrez la valeur du paramètre lam : ")
N = input("Entrez la valeur de N : ")

// Valeurs observées
Obs = grand(1, N, "poi", lam)
// Effectifs correspondants
v = tabul(Obs)

// Distribution théorique
m = max(v(:,1))
x = 1:m
y = exp(-lam)*lam^x ./ factorial(x)

// Tracé des diagrammes
clf()
bar(v(:,1)+.5, v(:,2)/N, width=0.4, 'red')
bar(x, y, width=0.4)
```

(on enregistre ce programme sous le nom `distrib_poisson.sce`*)*

II. Bilan sur la fonction grand

- Évaluer `help grand` et prendre connaissance des informations contenues dans la section `Générer des nombres aléatoires`. Compléter alors le texte ci-dessous.

`grand(1,N,"def")` permet de simuler `1` échantillon de `N` v.a.r. suivant la loi $\mathcal{U}([0,1[)$.

`grand(m,N,"def")` permet de simuler `m` échantillons de `N` v.a.r. suivant la loi $\mathcal{U}([0,1[)$.

`grand(1,N,"uin",a,b)` permet de simuler `1` échantillon de `N` v.a.r. suivant la loi $\mathcal{U}(\llbracket a,b\rrbracket)$.

`grand(m,N,"uin",a,b)` permet de simuler `m` échantillons de `N` v.a.r. suivant la loi $\mathcal{U}(\llbracket a,b\rrbracket)$.

`grand(1,N,"bin",n,p)` permet de simuler `1` échantillon de `N` v.a.r. suivant la loi $\mathcal{B}(\mathtt{n},\mathtt{p})$.

`grand(m,N,"bin",n,p)` permet de simuler `m` échantillons de `N` v.a.r. suivant la loi $\mathcal{B}(\mathtt{n},\mathtt{p})$.

`grand(1,N,"geom",p)` permet de simuler `1` échantillon de `N` v.a.r. suivant la loi $\mathcal{G}(p)$.

`grand(m,N,"geom",p)` permet de simuler `m` échantillons de `N` v.a.r. suivant la loi $\mathcal{G}(p)$.

`grand(1,N,"poi",lam)` permet de simuler `1` échantillon de `N` v.a.r. suivant la loi $\mathcal{P}(\mathtt{lam})$.

`grand(m,N,"poi",lam)` permet de simuler `m` échantillons de `N` v.a.r. suivant la loi $\mathcal{P}(\mathtt{lam})$.

`grand(1,N,"exp",1/alpha)` permet de simuler `1` échantillon de `N` v.a.r. suivant la loi $\mathcal{E}(\mathtt{alpha})$.

`grand(m,N,"exp",1/alpha)` permet de simuler `m` échantillons de `N` v.a.r. suivant la loi $\mathcal{E}(\mathtt{alpha})$.

Il faut noter que les v.a.r. de chaque échantillon sont **indépendantes**.

II.1. Simulation d'une v.a.r. suivant une loi normale

II.1.a) Utilisation de la fonction grand

▶ Si m, N, esp et sigma sont donnés, à quoi sert l'appel grand(m,N,"nor",esp,sigma) ?

La commande grand(m,N,"nor",esp,sigma) permet de simuler m échantillons de N v.a.r. indépendantes X_i suivant toutes la loi de Poisson de paramètre (esp,sigma).

II.1.b) Comparaison avec la distribution théorique

▶ Soit X une v.a.r. . Que signifie que $X \hookrightarrow \mathcal{N}(0,1)$? Détailler.

- $X(\Omega) =]-\infty, +\infty[$
- X admet pour densité la fonction φ définie par :

$$\varphi \ : \ \left| \begin{array}{ccc} \mathbb{R} & \to & \mathbb{R} \\ x & \mapsto & \dfrac{1}{\sqrt{2\pi}} \, \mathrm{e}^{-\frac{x^2}{2}} \end{array} \right.$$

▶ Écrire la fonction densiteNormaleCR, densité de la loi normale centrée réduite.

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

▶ On considère un vecteur G contenant N valeurs comprises entre -5 et 5.
On souhaite, grâce à histplot, générer le diagramme des fréquences de Obs. Comment le générer de sorte qu'il contienne 100 bâtons dont le premier commence en -5 et le dernier finit en 5 ?

On souhaite 100 écarts (pour définir 100 classes). Il faut donc que le vecteur définissant les classes contiennent 101 éléments : histplot(linspace(-5,5,101), Obs).

▶ Dans un nouvel onglet **SciNotes**, adapter les programmes précédents au cas de la loi normale centrée réduite.

```
// Valeur des paramètres
N = input("Entrez la valeur de N : ")

// Valeurs observées
Obs = grand(1, N, "nor", 0, 1)

// Tracé des diagrammes
clf()
histplot(linspace(-5,5,101), Obs)
plot(linspace(-5,5), densiteNormaleCR)

legend(["Diagramme distribution approchée","Densité"],"in_upper_right")
xtitle(["Loi normale centrée réduite :" ; "comparaison de la distribution
approchée et de la fonction densité"])
```