Nom :

Mathématiques approfondies
Cours ECG 2

Python - partie II

Thèmes

1. Algèbre bilinéaire
2. Compléments sur les variables aléatoires à densité
3. Convergence des variables aléatoires
4. Estimations
5. Compléments sur les fonctions de deux variables
6. Exercices

Chapitre

20. Compléments en statistiques

Lycée Saint Louis 2023/2024

Python - semestre II

```
NotesDS5=rd.randint(0,20,46)
```

C. FISZKA , *13 janvier 2024*

1 Algèbre bilinéaire

Exercice 1. ✦✦ **Méthode de la puissance itérée**

Dans la suite, A désigne une matrice de $\mathcal{M}_n(\mathbb{R})$ non nulle et diagonalisable. On pose λ, la plus grande valeur propre de A en valeur absolue.
Le but de cet exercice est de déterminer une approximation d'un vecteur propre de A associé à la valeur propre λ. Pour cela, on définit la suite de vecteurs $(X_k)_{k\in\mathbb{N}}$ par

$$X_0 \in \mathcal{M}_{n,1}(\mathbb{R}) \quad \text{et} \quad \forall\, k \in \mathbb{N}, \quad X_{k+1} = \frac{AX_k}{\|AX_k\|}$$

où $\|\cdot\|$ désigne la norme associée au produit scalaire canonique sur $\mathcal{M}_{n,1}(\mathbb{R})$. On admet que la suite $(X_k)_{k\in\mathbb{N}}$ converge vers un vecteur propre associé à λ, c'est-à-dire que, pour tout indice $i \in [\![1;n]\!]$, la suite de la i-ème coordonnée de X_k converge.

1. Écrire une fonction **norme** qui prend en argument une matrice colonne X et renvoie sa norme $\|X\|$.
2. Écrire une fonction **puissance(A, X0, k)** qui renvoie le terme X_k.
3. *Test.* On pose

$$A = \begin{bmatrix} 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix} \quad \text{et} \quad X_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

   Calculer (sans Python) les valeurs propres de A et donner un vecteur propre de norme 1 associé à la plus grande valeur propre. Comparer les résultats obtenus avec les résultats obtenus avec la fonction **puissance**.
4. Modifier le programme précédent pour construire une nouvelle fonction qui prend en arguments A, X_0, λ et affiche la plus petite valeur k telle que $\|AX_k - \lambda X_k\| \leqslant 10^{-6}$.
5. On suppose maintenant la matrice A inversible. Comment obtenir une approximation d'un vecteur propre d'une matrice associé à μ, la plus petite valeur propre de A en valeur absolue ?

Exercice 2. ✦✦ **Construction d'une famille orthonormée avec deux vecteurs**

1. Soient u, $v \in \mathbb{R}^n$ non colinéaires. Comment construire deux vecteurs e_1, $e_2 \in \mathbb{R}^n$ tels que

$$\|e_1\| = 1, \quad \|e_2\| = 1 \quad \text{et} \quad \text{Vect}(u,v) = \text{Vect}(e_1, e_2) \text{ ?}$$

2. **a)** Écrire une fonction `Scal` qui prend en arguments deux vecteurs de $\mathbb{R}^n$ et renvoie $\langle u, v\rangle$ où $\langle\cdot,\cdot\rangle$ correspond au produit scalaire canonique de $\mathbb{R}^n$.
   On identifiera un vecteur de $\mathbb{R}^n$ avec une matrice de $\mathcal{M}_{n,1}(\mathbb{R})$.

   b) En déduire un programme qui prend en arguments deux vecteurs de $\mathbb{R}^n$ puis teste si les deux vecteurs sont non colinéaires et, dans le cas où ils ne sont pas colinéaires, renvoie une base orthonormée du sous-espace vectoriel engendré par ces deux vecteurs.

2 Compléments sur les variables aléatoires à densité

2.1 Nouvelles méthodes de simulation

Exercice 3. ✦ **Simulation de la loi du χ^2**

Soient $n \in \mathbb{N}^*$ et $(X_i)_{i\in[\![1;n]\!]}$ un vecteur aléatoire de variables aléatoires mutuellement indépendantes et de même loi $\mathcal{N}(0;1)$. On dit que la somme $S_n = {X_1}^2 + \ldots + {X_n}^2$ suit une loi du χ^2 à n degrés de liberté.

1. Écrire un programme qui prend en argument n et simule la variable S_n.
   On pourra utiliser dans la bibliothèque ***numpy****, la commande* ***np.random.normal(mu,v,m)*** *pour simuler m réalisation d'une variable aléatoire suivant une loi normale $\mathcal{N}(\mu, v)$.*
2. En déduire un second programme qui prend en argument n et $m \in \mathbb{N}^*$ et renvoie un échantillon de taille m de S_n. Tracer les histogrammes.
3. Parmi ces trois densités, une seule correspond à la loi du χ^2 de degré 4, laquelle ?

$$f : x \in \mathbb{R} \mapsto \begin{cases} 0 & \text{si } x < 0 \\ \frac{1}{4}xe^{-\frac{x}{2}} & \text{si } x \geqslant 0, \end{cases} \qquad g : x \in \mathbb{R} \mapsto \frac{e^x}{(e^x+1)^2} \quad \text{et} \quad h : x \in \mathbb{R} \mapsto \begin{cases} 0 & \text{si } x < 0 \\ 4xe^{-2x} & \text{si } x \geqslant 0. \end{cases}$$

Exercice 4. ✦ **Simulation de la loi $\gamma(n)$**

1. Soit X une variable aléatoire suivant une loi uniforme sur $]0;1[$. Donner la loi de $Y = -\ln(X)$.
2. Soit $n \in \mathbb{N}^*$. En déduire un programme Python pour simuler une variable aléatoire de loi $\gamma(n)$.
3. Tester votre programme en affichant les histogrammes et en comparant avec une densité de $\gamma(n)$.

Exercice 5. ✦ **Simulation de la loi de Laplace**

Une variable aléatoire à densité X suit la loi de Laplace si une densité f est définie sur $\mathbb{R}$ par

$$f(x) = \frac{1}{2}e^{-|x|}.$$

1. Soit $U \hookrightarrow \mathcal{U}(]0;1])$. Donner la loi de $X = -\ln(U)$.
2. Justifier que si X et Y sont deux variables aléatoires définies sur un même espace probabilisé qui suivent une loi exponentielle alors la différence $X - Y$ suit une loi de Laplace.
3. En déduire un programme Python qui simule une loi de Laplace.
4. Tester votre programme en comparant les histogrammes obtenus avec la densité.

Exercice 6. ✦✦✦ **Simulation des lois normales par la méthode de Box-Müller**

- *La théorie*

  Soient X et Y deux variables aléatoires définies par

$$\begin{cases} X = \sqrt{W}\cos(\Theta) \\ Y = \sqrt{W}\sin(\Theta) \end{cases} \quad \text{où} \quad W \hookrightarrow \mathcal{E}(1/2) \quad \text{et} \quad \Theta \hookrightarrow \mathcal{U}([0;2\pi])$$

  avec W et Θ indépendantes. On admet que les variables X et Y sont indépendantes et de même loi $\mathcal{N}(0;1)$.

1. Soient U et V sont deux variables aléatoires indépendantes suivant la loi uniforme sur $]0;1]$. Justifier que les deux variables définies par
   $$\sqrt{-2\ln U}\cos(2\pi V) \quad \text{et} \quad \sqrt{-2\ln U}\sin(2\pi V)$$
   sont indépendantes et suivent une loi normale centrée réduite.

- *La pratique*

2. En déduire une fonction Python **NormaleCR** qui simule une loi normale centrée réduite.
3. Comment, à partir de **NormaleCR**, obtenir une simulation de la loi $\mathcal{N}(\mu;\sigma^2)$?
4. Conclure en donnant une fonction Python qui prend en argument $m \in \mathbb{N}^*$ et renvoie un échantillon de taille $2m$ de la loi normale centrée réduite avec l'histogramme de l'échantillon.

Exercice 7. ✦✦✦ **Simulation d'une loi de Poisson par des lois uniformes**

Soient $\lambda \in \mathbb{R}_*^+$ et $X_1, X_2, \ldots, X_n$ des variables aléatoires indépendantes, toutes de même loi exponentielle de paramètre 1. Soit N le plus grand entier n tel que $X_1 + \cdots + X_n < \lambda$, en convenant que $N = 0$ si $X_1 > \lambda$. Autrement dit, pour $n \in \mathbb{N}$,

$$[N = n] = [X_1 + \cdots + X_n < \lambda] \cap [X_1 + \cdots + X_n + X_{n+1} \geqslant \lambda].$$

- *La théorie*

1. Vérifier que pour tout $n \in \mathbb{N}$, $\mathbf{P}\big([\mathrm{N} \geqslant n]\big) = \mathbf{P}\big([\mathrm{N} \geqslant n+1]\big) + \frac{\lambda^n}{n!}\mathrm{e}^{-\lambda}$.
2. En déduire que N suit une loi de Poisson de paramètre λ.
3. On considère maintenant $(\mathrm{U}_n)_{n\in\mathbb{N}^*}$, une suite de variables aléatoires mutuellement indépendantes suivant toutes la loi uniforme sur $]0;1]$ et on définit la variable aléatoire X par

$$\forall\, \omega \in \Omega, \quad \mathrm{X}(\omega) = \begin{cases} 0 \text{ si } \mathrm{U}_1(\omega) < \mathrm{e}^{-\lambda} \\ \min\left\{n \in \mathbb{N}^* \;\middle|\; \mathrm{U}_1(\omega)\ldots\mathrm{U}_{n+1}(\omega) < \mathrm{e}^{-\lambda}\right\} \text{ sinon.} \end{cases}$$

Démontrer que X suit une loi de Poisson de paramètre λ.
Pour rappel, si $\mathrm{U} \hookrightarrow \mathscr{U}([0;1])$, *alors* $-\ln(\mathrm{U}) \hookrightarrow \mathscr{E}(1)$.

- *La pratique*

4. En déduire un programme qui simule la loi de Poisson de paramètre $\lambda \in \mathbb{R}_*^+$.

Exercice 8. ✦✦✦ **Simulations d'une loi bêta**

Soient $\alpha, \beta \in \mathbb{R}_*^+$. On pose $\displaystyle \mathrm{B}(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}\,\mathrm{d}t.$

On considère la fonction $f_{\alpha,\beta}$ définie sur $\mathbb{R}$ par $\displaystyle f_{\alpha,\beta}(x) = \frac{1}{\mathrm{B}(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}\,\mathbf{1}_{]0;1[}(x)$.

1. *La loi bêta*
   a) Vérifier que $f_{\alpha,\beta}$ est une densité de probabilité. On parle alors d'une loi bêta de paramètre (α,β).
   b) Écrire un programme qui prend en argument le paramètre (α,β) et renvoie la courbe de $f_{\alpha,\beta}$ sur $[0;1]$.
   On pourra utiliser la commande `sp.beta(alpha,beta)` après avoir importé la bibliothèque `scipy.special` *via* `sp`.
2. *Simulation pour des paramètres entiers - statistiques d'ordre et loi uniforme*
   Soient $n \in \mathbb{N}^*$ et $\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n$, n variables aléatoires indépendantes et de loi uniforme sur $[0;1]$.
   On admet l'existence de variables aléatoires à densité $\mathrm{Y}_1, \mathrm{Y}_2, \ldots, \mathrm{Y}_n$ telles que, pour tout ω de Ω, les réels $\mathrm{Y}_1(\omega)$, $\mathrm{Y}_2(\omega)$, …, $\mathrm{Y}_n(\omega)$ constituent un réarrangement par ordre croissant des réels $\mathrm{X}_1(\omega)$, $\mathrm{X}_2(\omega)$, …, $\mathrm{X}_n(\omega)$ de telle sorte que, pour tout ω de Ω :

$$\mathrm{Y}_1(\omega) \leqslant \mathrm{Y}_2(\omega) \leqslant \cdots \leqslant \mathrm{Y}_n(\omega).$$

   En particulier, $\mathrm{Y}_1 = \min(\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n)$ et $\mathrm{Y}_n = \max(\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n)$.
   a) Écrire un programme qui prend en arguments $n \in \mathbb{N}^*$ et $k \in [\![1;n]\!]$ et renvoie une simulation de Y_k.
   Indication. On pourra utiliser la commande `np.sort(A)` *qui prend en argument une matrice ligne et renvoie une matrice mais avec les coefficients de* `A` *ordonnés par ordre croissant.*
   b) En déduire un programme qui prend en arguments $n \in \mathbb{N}^*$ et $k \in [\![1;n]\!]$ et affiche l'histogramme associé à 5 000 réalisations de la variable Y_k ainsi que la courbe représentative de la fonction $f_{k,n+1-k}$. Tester et commenter.
3. *Simulation pour des paramètres supérieurs à 1 - méthode du rejet*
   Comment simuler une loi bêta de paramètres $(\alpha;\beta)$ par la méthode de rejet ?
4. *Simulation - cas général avec l'algorithme de Jönk*
   Soient $(\mathrm{U}_n)_{n\in\mathbb{N}^*}$ et $(\mathrm{V}_n)_{n\in\mathbb{N}^*}$ deux suites de variables aléatoires mutuellement indépendantes et suivant toutes la loi uniforme sur $]0;1]$. On note

$$\mathrm{N} = \inf\left\{n \in \mathbb{N}^* \,|\, {\mathrm{U}_n}^{1/\alpha} + {\mathrm{V}_n}^{1/\beta} \leqslant 1\right\}.$$

   On admet alors que $\frac{{\mathrm{U}_\mathrm{N}}^{1/\alpha}}{{\mathrm{U}_\mathrm{N}}^{1/\alpha} + {\mathrm{V}_\mathrm{N}}^{1/\beta}}$ suit une loi bêta de paramètres (α,β).
   En déduire un programme qui simule une loi bêta de paramètres (α,β).
5. *Estimations des paramètres*
   a) Écrire un programme Python qui prend en argument α, β et renvoie $\overline{x}$ (resp. v), la moyenne arithmétique (resp. la variance empirique) de 5000 réalisations d'une loi bêta de paramètre $(\alpha;\beta)$.

b) *Facultatif.* Donner la valeur exacte de l'espérance et la variance d'une loi bêta de paramètre (α, β).
On pourra admettre que pour tous $a, b \in \mathbb{R}_*^+$

$$B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \quad et \quad \Gamma(a+1) = a\Gamma(a).$$

6. Modifier le programme précédent en rajoutant le calcul des quantités suivantes. Tester et commenter.

$$\overline{x}\left(\frac{\overline{x}(1-\overline{x})}{v} - 1\right) \quad \text{et} \quad (1-\overline{x})\left(\frac{\overline{x}(1-\overline{x})}{v} - 1\right).$$

2.2 Exercices supplémentaires

Exercice 9. ✦ **Autour de la loi de Cauchy**

1. Soient X et Y deux variables aléatoires indépendantes suivant toutes les deux une loi normale centrée réduite.

a) En utilisant la commande `rd.normal(0,1)`, simuler la loi de Z = X/Y.

b) Comment obtenir un échantillon de Z contenant 10000 réalisations ? Calculer la moyenne empirique pour plusieurs échantillons. Commenter les résultats.

c) Tracer l'histogramme d'un échantillon issu de Z sur $[-10, 10]$ avec 10000 réalisations de Z et 30 classes. Superposer la courbe représentative de la fonction $x \mapsto \frac{1}{\pi(1+x^2)}$. Commenter.

2. On dit qu'une variable aléatoire suit une loi de Cauchy de paramètre $\lambda \in \mathbb{R}_*^+$, noté $X \hookrightarrow \mathscr{C}(\lambda)$ si une densité est donnée sur $\mathbb{R}$ par :

$$\forall x \in \mathbb{R}, \qquad f(t) = \frac{1}{\pi}\left(\frac{\lambda}{x^2+\lambda^2}\right).$$

a) Si $X \hookrightarrow \mathscr{C}(1)$, quelle est la loi de αX où $\alpha \in \mathbb{R}_*^+$? En déduire un programme qui permet de simuler une loi de Cauchy $\mathscr{C}(\lambda)$.

b) Soient X et Y, deux variables aléatoires indépendantes respectivement de loi $\mathscr{C}(\lambda)$ et $\mathscr{C}(\mu)$. On souhaite tester le résultat suivant :

$$X + Y \hookrightarrow \mathscr{C}(\lambda + \mu).$$

Proposer un script Python qui permet de tester ce résultat à l'aide d'histogrammes.

c) Soient $(X_i)_{i\in\mathbb{N}^*}$, une suite de variables aléatoires mutuellement indépendantes et toutes de loi $\mathscr{C}(1)$. Que dire de la loi de

$$\frac{1}{n}\sum_{i=1}^{n} X_i \quad ?$$

Exercice 10. ✦✦ **Autour de la loi de Fréchet**

Soit $s \in \mathbb{R}_*^+$. On dit qu'une variable aléatoire X suit une loi de Fréchet de paramètre s, noté $\mathscr{F}(s)$, si sa fonction de répartition est donnée par

$$\forall x \in \mathbb{R}, \quad F_s(x) = \begin{cases} \exp\left(-\left(\frac{x}{s}\right)^{-1}\right) & \text{si } x > 0 \\ 0 & \text{sinon.} \end{cases}$$

1. Démontrer que : Si $U \hookrightarrow \mathscr{U}([0,1])$ (loi uniforme continue) alors $s(-\ln(U))^{-1} \hookrightarrow \mathscr{F}(s)$.

2. Est-ce qu'une loi de Fréchet admet une espérance ? une variance ?

3. Écrire un programme qui prend en argument s et simule une loi de Fréchet de paramètre s.

4. *Loi d'un maximum de loi de Fréchet*
Soient $X_1, \ldots, X_n$, n variables aléatoires définies sur un même espace probabilisé, indépendantes et toutes suivantes une loi de Fréchet de paramètre s. On pose

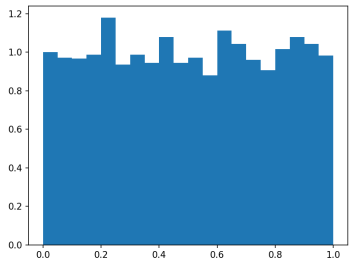
$$Y_n = \max(X_1, X_2, \ldots, X_n).$$

Écrire une fonction Python **MaxFrechet** qui prend en argument s, n et un entier naturel m et renvoie un échantillon de taille m de la variable Y_n.

5. À l'aide du code suivant et des résultats numériques. Conjecturer la loi de Y_n.

Editeur

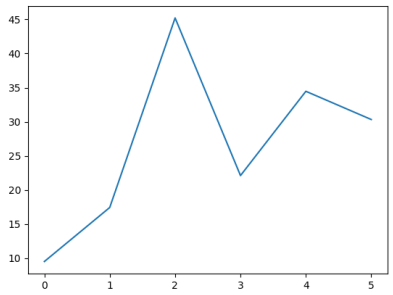
```
def Truc(s,n):
    plt.clf()
    Ech=MaxFrechet(s,n,5000)
    L=np.exp(-(Ech/(n*s))**(-1))

    plt.hist(L,20,density=True)
    plt.show()
```



6. Quelle propriété de la loi de Fréchet illustre le code et les résultats suivants ?

Editeur

```
L=[]

for p in range(2,8):
    S=0
    for k in range(1,10**p):
        S+=frechet(1)
    L.append(S/10**p)
plt.plot(L)
plt.show()
```




```
L=[9.52163, 17.43712, 45.20899, 22.11409,
    34.46171, 30.32867]
```

Exercice 11. ✦✦ **Exemple de question à l'oral d'HEC**

1. Commenter cette fonction Python.
2. Déterminer le graphe de sortie de la fonction `graph()`.

Editeur

```
def g(x):
    X=np.random.uniform(0,0.5,10000)
    Y=np.random.uniform(0,1/3,10000)
    S=0
    for k in range(10000):
        if X[k]+Y[k]<=x:
            S=S+1
    return S/10000
```

```
def graph():
    x=np.linspace(-0.1,1,101)
    Z=np.zeros(101)
    for k in range(101):
        Z[k]=g(x[k])
    plt.plot(x,Z)
    plt.show()
```

3 Convergence des variables aléatoires

3.1 Exemples et applications

Exercice 12. ✦✦ **Approche expérimentale du théorème limite central**

1. *Diagramme en bâtons et densité*
   Rappelons :
   - → Un diagramme en bâtons s'obtient par les commandes `plt.bar(x,y,width=1)` où `x` représente les abscisses et `y` les ordonnées.
   - → Le coefficient $\binom{n}{k}$ peut s'obtenir directement par `sp.binom(n,k)`.
   - → Pour limiter le graphe aux abscisses comprises entre `a` et `b`, on peut rajouter la ligne `plt.xlim(a,b)`.

a) Expliciter la loi de $S_n^* = \frac{X_n - np}{\sqrt{np(1-p)}}$ où $X_n \hookrightarrow \mathscr{B}(n;p)$.

b) Écrire une fonction qui prend en arguments $n \in \mathbb{N}^*$, $p \in]0;1[$ et affiche le diagramme en bâtons associé à la loi de S_n^*. C'est-à-dire que l'on place en abscisse les valeurs prises par la variable aléatoire S_n^* et en ordonnée les probabilités correspondantes.
On multipliera les ordonnées du diagramme en bâtons par $\sqrt{np(1-p)}$ afin que la sommes des aires des rectangles soit de 1.

c) Rajouter sur le graphique précédent la courbe représentative de $x \mapsto e^{-t^2/2}/\sqrt{2\pi}$.
On se limitera à l'intervalle $[-4;4]$.

d) Tester par exemple pour $n \in \{10;20;30;60\}$, $p \in \{0.3;0.5\}$ et commenter.

2. *Histogramme et densité*

a) Comment simuler la variable S_n^* ?

b) Écrire une fonction qui prend en arguments $n \in \mathbb{N}^*$, $p \in]0;1[$ et affiche sur le même graphe un histogramme de 10 000 réalisations de S_n^* et la courbe représentative de $x \mapsto e^{-t^2/2}/\sqrt{2\pi}$.
On pourra se restreindre à l'intervalle $[-4;4]$ *et considérer 30 classes.*

c) Tester pour $n \in \{500;2000;5000\}$, $p \in \{0.3;0.5\}$ et commenter.

Exercice 13. ✦✦ Comparaison des lois $\mathscr{B}\left(n;\frac{\lambda}{n}\right)$ et $\mathscr{P}(\lambda)$

1. *Comparaison des diagrammes en bâtons*

a) Écrire un programme qui prend en arguments $n \in \mathbb{N}^*$, $p \in]0;1[$ puis renvoie le diagramme en bâtons associé à la loi binomiale $\mathscr{B}(n;p)$.

b) Soient $\lambda \in]0,+\infty[$ et $X \hookrightarrow \mathscr{P}(\lambda)$.

i) Écrire un programme qui prend en arguments n, λ et renvoie la matrice ligne

$$\begin{bmatrix} p_0 & p_1 & \dots & p_n \end{bmatrix} \quad \text{où} \quad p_i = \frac{\lambda^i}{i!}e^{-\lambda}.$$

On pourra remarquer que $p_{i+1} = \lambda p_i/(i+1)$.

ii) En déduire un second programme qui prend en argument λ et n renvoie le diagramme en bâtons sur $[\![0;n_\lambda]\!]$ de la loi de Poisson $\mathscr{P}(\lambda)$.

c) Pour différentes valeurs de n et λ, superposer les diagrammes en bâtons associés aux lois $\mathscr{B}(n;\lambda/n)$ et $\mathscr{P}(\lambda)$. Commenter.

2. *Comparaison des histogrammes*
Sur un même graphique, placer les histogrammes issus de 5000 réalisations des lois $\mathscr{B}(n;\lambda/n)$ et $\mathscr{P}(\lambda)$. Tester pour $\lambda = 5$ et des valeurs de n de plus en plus grande. Commenter.

Exercice 14. ✦ Convergence et loi de Gumbel

On dit que X suit la loi de Gumbel si sa fonction de répartition F est définie par

$$\forall x \in \mathbb{R}, \qquad F(x) = e^{-e^{-x}}.$$

On considère aussi une suite de variables aléatoires indépendantes $(Y_n)_{n\in\mathbb{N}^*}$ suivant chacune la loi exponentielle $\mathscr{E}(1)$.

1. Écrire un programme qui prend en argument $n \in \mathbb{N}^*$ et simule la variable aléatoire

$$Z_n = \max(Y_1,\dots,Y_n) - \ln n.$$

2. Afficher sur un même graphique un histogramme associé à un échantillon de 1000 réalisations de Z_n et une densité de la loi de Gumbel. On pourra se restreindre à l'intervalle $[-2;7]$. Tester pour $n = 5$, 10, 50 ... Commenter.

Exercice 15. ✦ Simulation d'une loi normale par la méthode des douze uniformes

Le théorème limite central énonce que si $(X_n)_{n\in\mathbb{N}^*}$ est une suite de variables aléatoires mutuellement indépendantes suivant une loi uniforme sur $[0;1]$ alors

$$\left(\overline{X}_n{}^*\right)_{n\in\mathbb{N}^*} \xrightarrow[n\to+\infty]{\mathscr{L}} Z \quad \text{avec} \quad Z \hookrightarrow \mathscr{N}(0;1).$$

1. En se limitant à douze variables X_1, …, X_{12} suivant des lois uniformes, écrire une fonction Python qui renvoie une simulation approchée de la loi normale centrée réduite.

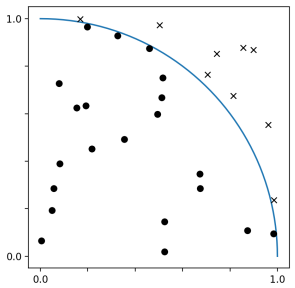
2. En déduire une seconde fonction qui prend en arguments μ, σ, m et renvoient m simulations d'une loi $\mathscr{N}(\mu,\sigma^2)$.

3. Tester votre programme en superposant sur une même graphique l'histogramme de 2000 simulations approchées de loi $\mathscr{N}(1;4)$ et une densité de cette loi.

3.2 Exemples de méthodes de Monte-Carlo

Les méthodes dites de Monte-Carlo sont toutes basées sur la loi (faible) des grands nombres.

Applications aux calculs d'aire



Commençons par un cas d'école : l'approximation de π.
On tire au hasard et uniformément un point dans le carré $[0;1]\times[0;1]$. La probabilité que le point soit situé dans le quart de disque est $\pi/4$ (aire du quart du disque $\pi\times 1^2/4$ sur l'aire du carré 1). Partant de ce constat, on peut simuler un grand nombre de tirages d'un point dans le carré et approximer la probabilité de $\pi/4$ par la fréquence empirique. En multipliant par 4, on obtient une approximation de π.
Ce qui donne ici :

Editeur

```
def approxPI(m):
# m correspond au nombre de tirages
    Compteur=0
    for i in range(m):
        x=rd.random()
        y=rd.random()
        if x**2+y**2<1:
            Compteur+=1
    print('Approximation:',4*Compteur/m)
```

Console

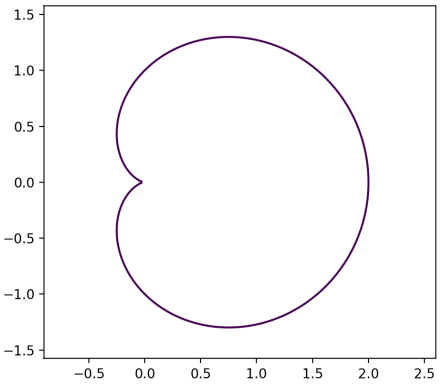
```
>>> approxPI(1000)   Approximation: 3.152
>>> approxPI(10000)  Approximation: 3.1412
>>> approxPI(50000)  Approximation: 3.14552
>>> approxPI(100000) Approximation: 3.14632

# à comparer à :
3.141592653589793 ...
```

La convergence est assez mauvaise mais il ne faut pas pour autant écarter la méthode. Elle s'avère par exemple particulièrement efficace en grande dimension.

Exercice 16. ✦✦ **Aire d'une cardioïde**

L'objectif de cet exercice est d'obtenir une approximation de l'aire de la partie délimitée par la courbe d'équation

$$\left(x^2+y^2-x\right)^2 = x^2+y^2.$$



1. Comment tirer un point au hasard dans le carré $[-0.5;2.5]\times[-1.5;1.5]$ en utilisant la commande **rd.random** ?
2. En déduire un programme qui tire au hasard un point dans le carré et déclare si le point est à l'intérieur de la cardioïde ou non.
3. À l'aide d'une méthode de Monte-Carlo, donner une approximation de l'aire de la cardioïde.
4. En remarquant que la courbe est la ligne de niveau L_0 d'une certaine fonction de deux variables, tracer la cardioïde.

Application à l'approximation d'intégrales

- **Principe**
  Considérons :
  - $g:[0,1]\to[a;b]$ une fonction continue dont on souhaite calculer l'intégrale $\displaystyle\int_0^1 g(t)\,\mathrm{d}t$.
  - $(U_i)_{i\in\mathbb{N}}$ une suite de variables aléatoires indépendantes suivant la loi uniforme sur $[0,1]$.
  - Pour tout $i\in\mathbb{N}$, on pose $g(U_i)$. D'après les lemme de l'égalité en loi et le lemme des coalitions, les variables sont indépendantes et de même loi.

Par le théorème de transfert, les variables $X_i = f(U_i)$ admettent toutes une même espérance donnée par

$$\mathbf{E}(X_i) = \int_0^1 g(t)\,\mathrm{d}t.$$

De même, ces variables admettent une variance et la loi faible des grands nombres donne

$$\frac{X_1+\cdots+X_n}{n} \xrightarrow[n\to+\infty]{\mathbf{P}} \mathbf{E}(X_1) = \int_0^1 g(t)\,\mathrm{d}t.$$

Dès lors, pour calculer une valeur approchée de l'intégrale, on peut simuler un grand nombre de fois les variables U_i, calculer les images $f(U_i)$ et en faire leur moyenne arithmétique.

Exercice 17. ✦✦

- *La théorie : estimation de la probabilité de l'erreur*
  Soit X, une variable aléatoire à valeurs dans $[a;b]$.

1. Pour quelle valeur de m, $\mathbf{E}\left((X-m)^2\right)$ atteint son minimum ? Avec $m=\frac{a+b}{2}$, déduire : $\mathbf{V}(X) \leqslant \dfrac{(b-a)^2}{4}$.
2. En reprenant les notations du début, justifier que

$$\forall \varepsilon \in \mathbb{R}_*^+, \quad \mathbf{P}\left(\left|\int_0^1 g(t)\mathrm{d}t - \frac{1}{n}\sum_{k=1}^{n} g(U_k)\right| > \varepsilon\right) \leqslant \frac{(b-a)^2}{4n\varepsilon^2} \quad (\bullet)$$

- *La pratique*

3. Calculer $I=\displaystyle\int_0^1 \frac{1}{1+t^2}\mathrm{d}t$. En déduire, une fonction Python qui prend en argument n qui permet d'approcher π.
4. Déterminer n afin d'obtenir une valeur à 10^{-3} près de π avec une probabilité d'au moins 95%. Commenter.

4 Estimations

Exercice 18. ✦ Soit $(X_1,\dots,X_n)$ un échantillon suivant la loi $\mathscr{P}(\lambda)$. Nous disposons des deux estimateurs de $\theta=\frac{1}{\lambda}$:

- La moyenne empirique $\overline{X_n}=\frac{1}{n}\sum_{i=1}^{n} X_i$;
- L'écart-type empirique $\sigma_n=\sqrt{\frac{1}{n-1}\sum_{k=1}^{n}\left(X_k-\overline{X_n}\right)^2}$.

On souhaite comparer ces deux estimateurs.

1. Écrire un programme qui prend en argument n et λ et simule $\overline{X_n}$, puis σ_n.
2. Afficher des histogrammes et comparer les estimateurs.

Exercice 19. ✦ Soient $n\in\mathbb{N}^*$ et $(X_1,X_2,\dots,X_n)$ un échantillon d'une loi de Poisson $\mathscr{P}(\theta)$. On cherche à estimer $\exp(-\theta)$. Pour cela, on pose :

$$A_n=\left(1-\frac{1}{n}\right)^{\sum_{i=1}^{n} X_i} \quad \text{et} \quad B_n=\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}_{[X_i=0]}.$$

1. Écrire deux programmes qui prend en arguments n, θ et simulent respectivement A_n et B_n.
2. On suppose dans la suite que $\theta=1$.
   En déduire un programme qui prend en argument n et renvoie une approximation de l'espérance de A_n et B_n.
   Que peut-on conjecturer sur le biais de chacun de ces estimateurs ?
3. Afficher l'histogramme de 20000 réalisations de A_{100} et B_{100}. Quel est le meilleur estimateur de $\exp(-\theta)$?
   On pourra considérer les classes : `classe=np.linspace(0.2,0.5,20)`.

Exercice 20. ✦✦ Soit $(X_n)_{n\in\mathbb{N}^*}$ une suite de variables indépendantes de même loi uniforme discrète sur $[\![1;N]\!]$. Pour tout $n\in\mathbb{N}^*$ et $k\in[\![1;N]\!]$, on pose :

- Y_n le nombre de valeurs distinctes prises par $X_1,\dots,X_n$.
- $Z_{k,n}$ égale à 1 si au moins l'un des X_i pour $i\in[\![1;n]\!]$ prend la valeur k et 0 sinon.

1. *La pratique*
   a) Écrire un programme qui prend en arguments n, N et simule le vecteur aléatoire

$$\begin{bmatrix} Z_{1,n} & Z_{2,n} & \dots & Z_{N-1,n} & Z_{N,n}\end{bmatrix}.$$

   b) En déduire un programme qui prend en arguments n, N et simule Y_n.
   c) Vérifier numériquement que Y_n est un estimateur asymptotiquement sans biais de N.
   Un estimateur est asymptotiquement sans biais si son biais b_n tend vers 0 lorsque $n\to+\infty$.
2. *La théorie*
   a) Justifier que pour tout $k\in[\![1;N]\!]$, $\mathbf{E}(Z_k)=1-\left(1-\frac{1}{N}\right)^n$.

b) En déduire l'espérance de Y_n et le fait que c'est un estimateur sans biais.

c) Montrer en plus que ce dernier est un estimateur convergent de N.

Exercice 21. ✦✦ **Comparaison des intervalles de confiance - cas du paramètre d'une loi de Bernoulli**

Soit $(X_i)_{i\in\mathbb{N}^*}$, une suite de variables aléatoires indépendantes et de même loi de Bernoulli $\mathscr{B}(p)$. Pour les tests, on choisit $p = 1/3$, $n = 1000$, $\alpha = 0.05$.

1. Écrire un programme `ech` qui prend en arguments n, p et renvoie une réalisation de $(X_1,\dots,X_n)$.

- *Estimation à partir de l'inégalité de Bienaymé-Tchebychev*
  On rappelle que l'intervalle de confiance de p à un niveau de confiance $1-\alpha$ est donné par :
$$\left[\overline{X}_n - \frac{1}{2\sqrt{n\alpha}} ; \overline{X}_n + \frac{1}{2\sqrt{n\alpha}}\right].$$

2. **a)** Écrire un programme Python qui prend en arguments n, p, α puis :
- → Crée un échantillon de taille n ;
- → Calcule l'intervalle de confiance associé à l'échantillon ;
- → Renvoie 1 si le paramètre p est bien dans l'intervalle, 0 sinon.

b) Afficher ensuite une approximation de la probabilité que le paramètre p soit bien dans l'intervalle de confiance calculé ainsi que la longueur de l'intervalle de confiance.

- *Intervalle de confiance asymptotique du paramètre d'une loi de Bernoulli*
  On a démontré qu'un intervalle de confiance asymptotique de p au niveau de confiance $1-\alpha$ est donné par
$$\left[\overline{X_n} - \frac{t_\alpha}{2\sqrt{n}} ; \overline{X_n} + \frac{t_\alpha}{2\sqrt{n}}\right]$$
  où t_α est l'unique solution de $\Phi(t_\alpha) = 1 - \frac{\alpha}{2}$ avec Φ la fonction de répartition de la loi normale centrée réduite.

3. Reprendre les deux questions précédentes avec ce nouvel intervalle.

4. Comparer et commenter les résultats obtenus.

Exercice 22. ✦ **Estimation par intervalle de confiance de la moyenne d'une loi normale d'écart type connu**

Dans une population donnée, une étude statistique faite sur un groupe de 100 personnes donne lieu à la série statistique suivante.

Poids	48	49	50	51	52	53	54	55
Effectif	3	5	2	6	6	10	12	10
Poids	56	57	58	59	60	61	62	63
Effectif	9	8	8	6	5	4	3	3

On suppose que le poids d'un individu du groupe est une variable aléatoire X qui suit une loi normale d'écart-type $\sigma = 3,5$. Dans chaque groupe de 100 personnes étudié, on désigne par X_i la variable aléatoire égale au poids du i-ème individu, pour tout $i \in [\![1;100]\!]$.

1. Calculer l'intervalle de confiance obtenu par l'inégalité de Bienaymé-Tchebychev.

2. En utilisant les propriétés de stabilité des lois normales, on montre qu'un intervalle de confiance de niveau $1-\alpha$ est donné par
$$\left[\overline{X}_n - t_\alpha\frac{\sigma}{\sqrt{n}} ; \overline{X}_n + t_\alpha\frac{\sigma}{\sqrt{n}}\right].$$
Expliciter cet intervalle avec les données obtenues.

3. Comparer les deux méthodes.

5 Compléments sur les fonctions de deux variables

Exercice 23. ✧ La fonction définie sur $\left(\mathbb{R}_+^*\right)^2$ par $f(x,y) = x^y - y^x$ possède (e, e) comme point critique. Retrouver graphiquement sa nature.

Exercice 24. ✦ **Points fixes dégénérés**

On définit les fonctions f, g et h sur $\mathbb{R}^2$ par :
$$f(x,y) = x^2 + y^3 \quad \text{et} \quad g(x,y) = x^3 + xy^2 - x^2y - y^3, \quad h(x,y) = x^4 + y^3 - 3y - 2.$$

1. Vérifier que dans les deux cas $(0;0)$ est un point critique.

2. Tracer les surfaces représentatives et conjecturer la nature du point critique $(0;0)$.

6 Exercices supplémentaires et sujets

Exercice 25. ✦ Soit $\mathscr{F} = (f_1, f_2, \ldots, f_n)$ une famille de n vecteurs de $\mathbb{R}^n$. On note P la matrice de la famille $\mathscr{F}$ dans la base canonique. Proposer un programme qui prend en argument la matrice P et indique si $\mathscr{F}$ est une base orthonormée ou non.

Exercice 26. ✦✦ Soit $p \in \mathbb{N}^*$.

1. Quelle est la loi suivie par la variable N ?
2. Préciser $X(\Omega)$. Calculer pour tout couple d'entiers (i,k) la probabilité $\mathbf{P}_{(N=k)}(X=i)$. En déduire la loi de X (on ne cherchera pas à simplifier la somme obtenue).
3. Calculer $\mathbf{E}(X)$

Editeur

```
import numpy.random as rd
def X(p):
    N=1+np.floor(p*rd.rand())
    # floor(.) renvoie la partie entière
    x=0
    i=0
    while i <N:
        x=x+(rd.random()<0.5)
        i+=1
    return x
```

Exercice 27. ✦ Quelle loi est simulée par le code suivant ?

Editeur

```
import numpy as np

def Mystere():
    u=np.random.rand(4)
    return -np.log(np.prod(u))
```

Exercice 28. ✦ **Retour sur la méthode de Monte-Carlo**
Soient $(U_i)_{i\in\mathbb{N}^*}$ des variables aléatoires réelles indépendantes de loi uniforme sur $[0;1]$ et N une variable aléatoire de loi géométrique de paramètre p indépendante de la suite $(U_i)_{i\in\mathbb{N}^*}$. On pose

$$X = \max_{1\leqslant i\leqslant N} U_i.$$

1. Déterminer la fonction de répartition de la variable aléatoire X.
2. Calculer l'espérance de X.
3. Simuler la variable et vérifier votre résultat.

Exercice 29. ✦✦ **Exemple de marche aléatoire asymétrique**
Soit $(X_n)_{n\in\mathbb{N}^*}$ une suite de variables aléatoires mutuellement indépendantes telles que pour tout $n \in \mathbb{N}^*$,

$$\mathbf{P}(X_n = 1) = p \quad \text{et} \quad \mathbf{P}(X_n = -1) = 1-p.$$

1. Écrire une fonction prenant en argument p et simule la variable aléatoire X_1.
2. On pose $S_0 = 0$ et pour tout $n \in \mathbb{N}^*$, $S_n = X_1 + X_2 + \cdots + X_n$.
   Écrire une fonction d'arguments p et n qui renvoie la liste $[S_0, S_1, \cdots, S_n]$.
3. Conjecturer la limite de la suite (S_n) en fonction de p.
4. Soit $n \in \mathbb{N}^*$. On note $T_n = \min\{k \in \mathbb{N} : |S_k| = n\}$. Écrire une fonction d'arguments p, n qui renvoie une simulation de la variable aléatoire T_n.

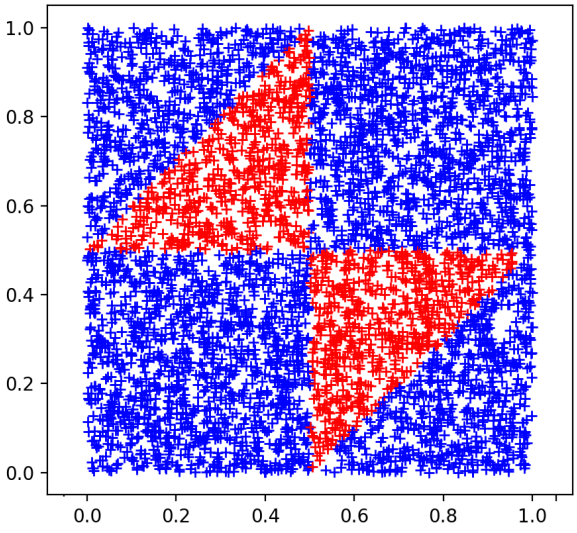
Exercice 30. ✦✦ On coupe un morceau de bois de longueur 1 en trois morceaux.

1. Donner un exemple dans lequel il n'est pas possible de former un triangle.
2. *Dans la suite, on admet que pour obtenir un triangle (non plat), les inégalités suivantes doivent être vérifiées simultanément :*
   $$a < b+c, \quad b < a+c, \quad c < a+b$$
   où a, b et c sont les longueurs des trois morceaux obtenus.

   On écrit le script suivant en Python et on affiche le résultat obtenu. Que peut-on conjecturer sur la probabilité d'avoir un triangle à partir des trois morceaux coupés aléatoirement ?

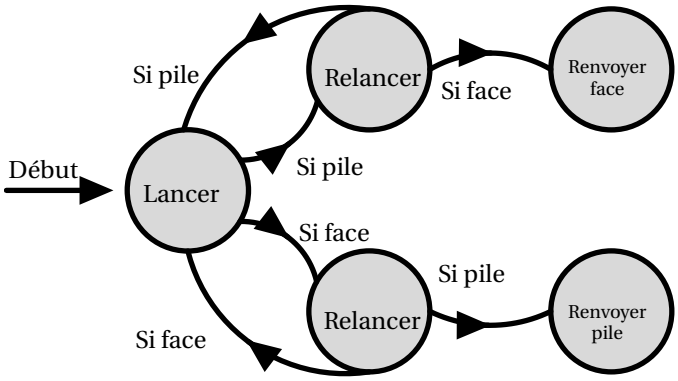
Editeur

```
m=5000
for i in range(m):
    x=np.random.rand()
    y=np.random.rand()
    L1=min(x,y)
    L2=max(x,y)-min(x,y)
    L3=1-max(x,y)
    if L1<L2+L3 and L2<L1+L3 and L3<L1+L2:
        plt.plot(x,y,'r+')
    # permet d'afficher une croix rouge
    else:
        plt.plot(x,y,'b+')
    # permet d'afficher une croix bleue
plt.axis('equal')

plt.show()
```



Exercice 31. ✦ **Pile - Face**

Le but de cet exercice est d'étudier un algorithme permettant de générer un pile ou face équilibré à partir d'une pièce truquée. On suppose que la pièce (truquée) renvoie pile avec probabilité $p \in]0,1[$ avec $p \neq 1/2$. Voici une description schématique de l'algorithme. On se place dans le cas où les lancers successifs de la pièce sont mutuellement indépendants.



1. Écrire une fonction `Bernoulli` qui prend en argument $p \in [0;1]$ et simule une variable aléatoire de loi $\mathscr{B}(p)$.
2. Donner une fonction Python `algo` qui renvoie pile (1) ou face (0) en suivant l'algorithme décrit ci-dessus.
3. Effectuer 5000 réalisations et afficher les fréquences d'obtention de pile et face. Commenter.
4. Modifier la fonction `algo` pour afficher en plus le nombre de lancers nécessaires pour que l'algorithme se termine.
5. On note T la variable aléatoire donnant le nombre de lancers nécessaires pour que l'algorithme se termine. Estimer l'espérance **E**(T).

Exercice 32. ✦✦ **Le problème du collectionneur** *d'après HEC 2022*

Le but de ce problème est de mettre en évidence quelques résultats asymptotiques liés au modèle du collectionneur de vignettes. Dans chaque paquet de céréales se trouve une vignette et il y a en tout des vignettes de n types différents, où n est un entier supérieur ou égal à 1. Chacun des n types de vignettes se retrouve avec la même fréquence dans les paquets de céréales. Une collection est alors complète lorsqu'elle comporte n vignettes de types différents.

On modélise le nombre total de paquets de céréales qu'il est nécessaire d'acheter pour obtenir la collection complète de n vignettes de types différents par la variable aléatoire notée C_n.

On pose par convention $C_0 = 0$ et pour tout entier $i \in [\![1;n]\!]$, on note C_i le nombre d'achats de paquets de céréales nécessaires pour obtenir i vignettes de types différents.

De même, pour tout $i \in [\![1;n]\!]$, on pose $X_i = C_i - C_{i-1}$, qui représente le nombre d'achats supplémentaires de paquets de céréales qu'il est nécessaire d'effectuer pour obtenir une nouvelle vignette d'un type différent des $(i-1)$ vignettes de types différents déjà obtenues. Par convention, on pose $X_1 = C_1 = 1$.

On suppose que les variables aléatoires $X_1, \ldots, X_n$ sont mutuellement indépendantes. Enfin, on pose :

$$V_n = \frac{C_n}{n} - \ln(n)$$

On admet que $C_n = X_1 + \cdots + X_n = \sum_{i=1}^{n} X_i$ et pour tout $i \in [\![1;n]\!]$, la variable aléatoire X_i suit la loi géométrique $\mathscr{G}\left(\frac{n-i+1}{n}\right)$.

1. Écrire une fonction python qui prend en argument $p \in]0;1[$ et simule un loi géométrique de paramètre p. On rappelle que la commande `np.random.rand()<p` permet de simuler une loi de Bernoulli de paramètre p.
2. Écrire une fonction d'en-tête `def simulV(n)` qui pour un entier n fourni en entrée, renvoie une simulation de la variable aléatoire V_n définie en introduction de ce problème.

3. À la suite de la fonction `simulV`, écrire une fonction `simulVech` qui construit un vecteur-ligne V contenant 1000 simulations indépendantes de la variable aléatoire V_n pour un certain entier n entré par l'utilisateur.
4. On complète ce programme par le code suivant. Tester le pour $n = 5$, $n = 10$, $n = 50$ et $n = 100$.

Editeur

```
import matplotlib.pyplot as plt

n=100

def f(x):
    return np.exp(-x-np.exp(-x))

absc=np.linspace(-2,10,100)
y=f(absc)

plt.clf()
plt.plot(absc,y)
plt.hist(simulVech(n),20,density=True)
plt.show()
```

5. Que peut-on observer sur ces figures ? Quelle conjecture peut-on en déduire pour la suite $(V_n)_{n\in\mathbb{N}^*}$?

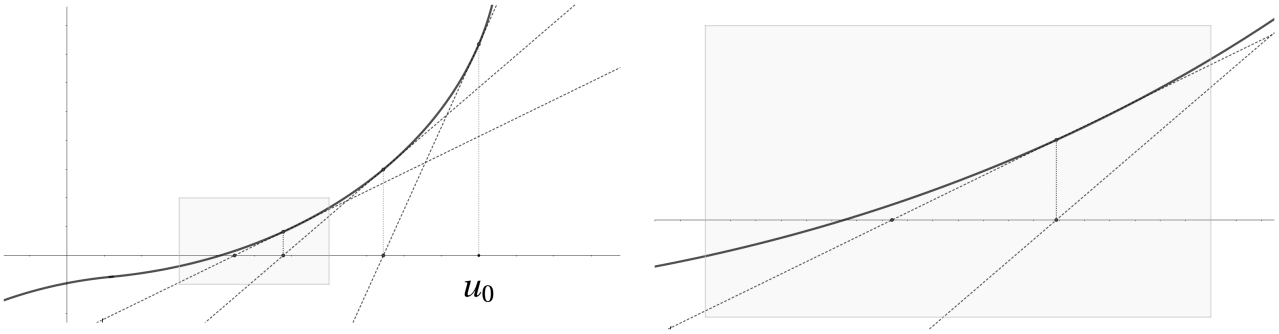
Exercice 33. ✦✦ **La méthode de Newton**

Partie A - Étude théorique

Soit $f : [a;b] \to \mathbb{R}$ de classe $\mathscr{C}^1$. On suppose que f' ne s'annule pas et que f s'annule une unique fois en un point c. On construit la suite u de la façon suivante :

- On part de $u_0 = a$.
- Pour tout $n \in \mathbb{N}$, u_{n+1} est l'abscisse du point d'intersection de l'axe des abscisses et de la tangente à la courbe représentative de f au point d'abscisse u_n.

1. Ci-dessous la courbe de f et des tangentes. Placer u_1, u_2 et u_3.
Conjecturer le comportement limite de la suite $(u_n)_{n\in\mathbb{N}}$.



2. Écrire l'équation de la tangente au point d'abscisse u_n. Vérifier que pour tout $n \in \mathbb{N}$,

$$u_{n+1} = u_n - \frac{f(u_n)}{f'(u_n)}.$$

3. Si la suite u converge vers une limite finie, justifier que u converge vers c.
4. Dans cette question, on prend l'exemple de $f : x \in [1;2] \mapsto x^2 - 2$.

   a) Vérifier que pour tout $n \in \mathbb{N}$, $\quad u_{n+1} - \sqrt{2} = \dfrac{\left(u_n - \sqrt{2}\right)^2}{2u_n}$.

   Puis, $\qquad \left|u_{n+1} - \sqrt{2}\right| \leqslant \left|u_n - \sqrt{2}\right|^2 \quad$ et $\quad \left|u_n - \sqrt{2}\right| \leqslant |u_0 - \sqrt{2}|^{2^n}$.

   b) En remarquant que $\left|u_0 - \sqrt{2}\right| \leqslant 1/2$, démontrer la convergence de la suite u vers $\sqrt{2}$.

Partie B - Python

5. Écrire un programme qui prend en entrée un entier n, u_0, f et f' et renvoie u_n.

6. Déduire de la question 4. une fonction Python qui prend en argument une précision p et renvoie une approximation de $\sqrt{2}$ à p-près.

Partie C - En dimension n

Soit F une fonction de $\mathbb{R}^n$ à valeurs dans $\mathbb{R}^n$. Une telle fonction est définie par ses m fonctions composantes à valeurs réelles

$$\mathrm{F} : \boldsymbol{x} \in \mathbb{R}^n \mapsto \big(\; f_1(\boldsymbol{x}), \quad f_2(\boldsymbol{x}), \quad \ldots, \quad f_n(\boldsymbol{x}) \; \big) \in \mathbb{R}^n.$$

On suppose dans la suite que toutes les fonctions composantes f_i sont de classe $\mathscr{C}^1$ sur $\mathbb{R}^n$ et on définit la matrice jacobienne de F par :

$$\mathrm{J}(\boldsymbol{x}) = \begin{bmatrix} \partial_1 f_1(\boldsymbol{x}) & \partial_2 f_1(\boldsymbol{x}) & \cdots & \partial_n f_1(\boldsymbol{x}) \\ \partial_1 f_2(\boldsymbol{x}) & \partial_2 f_2(\boldsymbol{x}) & \cdots & \partial_n f_2(\boldsymbol{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \partial_1 f_n(\boldsymbol{x}) & \partial_2 f_n(\boldsymbol{x}) & \cdots & \partial_n f_n(\boldsymbol{x}) \end{bmatrix}.$$

La méthode de Newton se généralise à plusieurs dimensions pour résoudre l'équation

$$\mathrm{F}(\boldsymbol{x}) = 0_{\mathbb{R}^n} \quad \text{d'inconnue } \boldsymbol{x} \in \mathbb{R}^n \qquad (\bullet)$$

Lorsque la matrice Jacobienne est inversible, on définit la suite $(\boldsymbol{x}_i)_{i\in\mathbb{N}}$ par $\boldsymbol{x}_0$ et

$$\forall\, i \in \mathbb{N}, \quad \boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \mathrm{J}(\boldsymbol{x}_i)^{-1}\mathrm{F}(\boldsymbol{x}_i)$$

où on identifie $\mathscr{M}_{n,1}(\mathbb{R})$ avec $\mathbb{R}^n$. Sous certaines conditions, on montre que la suite $(\boldsymbol{x}_i)_{i\in\mathbb{N}}$ converge vers une solution de l'équation $(\bullet)$.

Exemple
Considérons le système

$$\begin{cases} \cos(x) &= \sin(y) \\ \mathrm{e}^{-x} &= \cos(y) \end{cases} \quad \text{d'inconnue } (x,y) \in \mathbb{R}^2 \qquad (\star)$$

1. Déterminer une fonction $\mathrm{F} : \mathbb{R}^2 \to \mathbb{R}^2$ de sorte que l'équation $(\star)$ soit équivalente à $\mathrm{F}(x,y) = (0,0)$. Préciser la matrice jacobienne de F. En déduire deux programmes, un pour le calcul de $\mathrm{F}(x,y)$, un autre pour $\mathrm{J}(x,y)$.

2. En déduire un programme et une approximation d'une solution de $(\star)$.

La méthode de Newton est assez efficace et permet une convergence rapide vers une solution. Cette méthode a de nombreuses variantes et extensions. Citons par exemple, son application dans les « théorèmes K.A.M » dont l'étude de la stabilité du système solaire.

CHAPITRE 20

Statistiques

La statistique est aujourd'hui un fait social total : elle règne sur la société, régente les institutions et domine la politique. Un vêtement de courbes, d'indices, de graphiques, de taux recouvre l'ensemble de la vie. L'éducation disparaît derrière les enquêtes PISA, l'université derrière le classement de Shanghai, les chômeurs derrière la courbe du chômage... La statistique devait refléter l'état du monde, le monde est devenu un reflet de la statistique.

OLIVIER REY
Mathématicien et philosophe contemporain

1 Rappels en statistiques univariées

Soit Ω un ensemble.

- Ω désigne **la population** qui fera l'objet de l'étude.
- Les éléments $\omega_i \in \Omega$ sont les **individus**.
- Une **variable statistique** (ou **caractère**) X sur la population Ω est une application $X : \Omega \to F$. Dans le cas où F est une partie de $\mathbb{R}$, on dit que le caractère est **quantitatif**. Sinon, on dit que le caractère est qualitatif.
- Si Ω est un ensemble fini, le nombre d'éléments de Ω, noté $\text{Card}(\Omega)$ est **l'effectif de la population**.
- Quand il n'est pas possible d'étudier chaque individu de la population, on étudie seulement les individus d'une partie finie E de la population Ω. Dans ce cas, la partie E est appelée **échantillon**. Le nombre d'individus de l'échantillon, Card(E), est **la taille de l'échantillon**. Ainsi, si N désigne la taille, on peut écrire $E = \{e_1, e_2, \cdots, e_N\}$ où les e_i sont distincts deux à deux.

On distinguera deux types de variables statistiques.

- Si l'ensemble des valeurs prises par la variable, noté $X(\Omega)$, est un ensemble fini, on dit que la variable quantitative est **discrète**.
- Dans le cas contraire, on dit que la variable quantitative est **continue.**

1.1 Cas discret

Soit X une variable statistique discrète et $E = \{e_1, e_2, \cdots, e_N\}$ un échantillon.

- La donnée du N-uplet des observations $x = \big(X(e_1), X(e_2), \cdots, X(e_N)\big)$ définit une **série statistique**.
- L'échantillon étant un ensemble fini, l'ensemble des valeurs prises par la variable X sur E est aussi fini. On peut l'écrire
$$X(E) = \{m_1, m_2, \cdots, m_p\} \quad \text{où} \quad m_1 < m_2 < \cdots < m_p$$
où chaque m_i est une **valeur** ou **modalité.**
- **L'effectif d'une modalité** m_i est le nombre d'individu de E pour lequel le caractère prend la modalité m_i. C'est-à-dire
$$n_i = \text{Card}\big\{e \in E \mid X(e) = m_i\big\}.$$
Si N est la taille de l'échantillon, $N = \sum_{i=1}^{p} n_i$.

- **La fréquence d'une modalité** m_i est la quantité

$$f_i = \frac{\text{Effectif de la modalité}}{\text{Taille de l'échantillon}} = \frac{n_i}{\text{N}}.$$

On a alors $\sum_{i=1}^{p} f_i = 1$.

- **La fréquence cumulée d'une modalité** m_i est la somme de toutes les fréquences des modalités qui lui sont inférieures. Autrement dit,

$$\text{F}_i = \sum_{m_j \leqslant m_i} f_j.$$

Remarque. Donner une série statistique est équivalent à la donnée du couple (m, n) où

- $m = (m_1, m_2, \cdots, m_p)$ est le p-uplet constitué des modalités,
- $n = (n_1, n_2, \cdots, n_p)$ est le p-uplet constitué des effectifs.

Exemple. Le site de L'I.N.S.E.E (Institut national de la statistique et des études économiques) contient de très nombreuses statistiques en libre accès. Voici le début de la liste du nombre d'habitants par département.

Estimations de population par sexe et âge au 1er janvier 2022 : comparaisons départementales

	Ensemble	Part des femmes (en %)	Part des hommes (en %)	Part des 0 à 24 ans (en %)	Part des 25 à 59 ans (en %)	Part des 60 ans ou plus (en %)	dont part des 75 ans ou plus (en %)
Ain	665 391	50,7	49,3	30,5	44,7	24,8	*8,6*
Aisne	524 403	51,2	48,8	29,8	41,6	28,6	*9,7*
Allier	331 757	52,0	48,0	24,9	38,9	36,1	*14,4*
Alpes-de-Haute-Provence	165 582	51,4	48,6	24,6	40,2	35,2	*13,4*
Hautes-Alpes	141 059	51,2	48,8	24,8	41,3	33,9	*12,3*
Alpes-Maritimes	1 103 555	52,8	47,2	26,3	42,4	31,2	*13,0*
Ardèche	330 865	51,2	48,8	25,8	40,8	33,5	*12,4*
Ardennes	265 285	51,1	48,9	27,8	42,0	30,3	*10,4*
Ariège	153 126	51,1	48,9	24,3	40,7	35,1	*13,4*
Aube	311 083	51,4	48,6	29,9	41,5	28,6	*10,4*

Ci-dessous, la seconde colonne exportée en Python dans une liste.

```
L=[665391,
524403,
331757,
165582,
141059,
1103555,
330865,
265285,

...

294436,
868846,
299348]
```

Par exemple, on peut y lire que le département Nord est le plus peuplé (au 1er janvier 2022) avec 2 606 873 habitants.

1.2 Cas continu

On découpe l'ensemble des valeurs possibles $\text{X}(\Omega)$ en un certain nombre d'intervalles. Notons $p \in \mathbb{N}^*$ le nombre d'intervalles choisis. Ces intervalles doivent être deux à deux disjoints, et leur réunion est égale à (ou contient) l'ensemble des valeurs possibles. Plus précisément, en notant $a_1, a_2, \ldots, a_{p+1} \in \mathbb{R}$, les bornes de tous ces intervalles avec $a_1 < a_2 < \cdots < a_{p+1}$, on a

$$\text{X}(\Omega) \subset \bigcup_{i=1}^{p} [a_i ; a_{i+1}[.$$

L'intervalle $[a_i ; a_{i+1}[$ est une **classe**.

Remarques.

- On peut ainsi définir l'effectif d'une classe et sa fréquence.

- On peut utiliser ces définitions pour une variable discrète lorsque l'ensemble des valeurs prises est trop important. Par exemple, pour étudier le nombre d'habitants par département, on pourra faire des tranches de 200 000 habitants.

2 Paramètres

2.1 Paramètres de position

DÉFINITION — **La moyenne empirique**

*Soit $x = (x_i)_{1\leqslant i\leqslant \mathrm{N}}$, une série statistique. On définit la **moyenne** de la série par*

$$\overline{x} = \frac{1}{\mathrm{N}}\sum_{i=1}^{\mathrm{N}} x_i.$$

Si $(m_i)_{1\leqslant i\leqslant p}$, $(n_i)_{1\leqslant i\leqslant p}$ et $(f_i)_{1\leqslant i\leqslant p}$ désignent respectivement les modalités, les effectifs et les fréquences, on a aussi

$$\overline{x} = \frac{1}{\mathrm{N}}\sum_{i=1}^{p} n_i\, m_i = \sum_{i=1}^{p} f_i\, m_i$$

Exemple. Avec Python, la moyenne est obtenue par la commande `mean()` qui prend en argument une liste.
Ci-contre, le calcul du nombre d'habitants moyen par département.

Console

```
>>> import numpy as np
>>> np.mean(L)
671419.7623762377
```

DÉFINITION — **La médiane d'une série statistique - (hors-programme)**

*Soit $x = (x_i)_{1\leqslant i\leqslant \mathrm{N}}$ une série statistique ordonnées suivant l'ordre croissant. On définit la **médiane** de la série par*

- x_p *si* N *est impair avec* $p = (\mathrm{N}+1)/2$,
- $\frac{x_p + x_{p+1}}{2}$ *si* N *est pair avec* $p = \mathrm{N}/2$.

Autrement dit, la médiane un nombre réel m_e tel que le nombre d'individus pour lesquels X prend une valeur inférieure ou égale à m_e soit égal au nombre d'individus pour lesquels X prend une valeur supérieure ou égale à m_e.

Exemple. Avec Python, la médiane est obtenue par la commande `median()` qui prend en argument une liste.

Console

```
>>> import numpy as np
>>> np.median(L)
524506.0
```

2.2 Paramètres de dispersion

DÉFINITIONS — **La variance empirique et l'écart-type**

Soit x, une série statistique.

- ***La variance** de la série est le réel, noté $\sigma(x)^2$, défini par*

$$\sigma(x)^2 = \frac{1}{\mathrm{N}}\left((x_1-\overline{x})^2 + (x_2-\overline{x})^2 + \ldots + (x_{\mathrm{N}}-\overline{x})^2\right) = \frac{1}{\mathrm{N}}\sum_{i=1}^{\mathrm{N}}(x_i-\overline{x})^2.$$

- ***L'écart-type**, $\sigma(x)$ d'une série statistique est défini comme la racine carrée de la variance.*

Remarques.

- Si les modes $m_1, m_2, \cdots, m_p$ de la série sont donnés avec leurs effectifs $n_1, n_2, \cdots, n_p$ ou leurs fréquences $f_1, f_2, \ldots, f_p$,

alors

$$\sigma(x)^2 = \frac{1}{N}\sum_{i=1}^{p} n_i\left(m_i - \overline{x}\right)^2 = \sum_{i=1}^{p} f_i\left(m_i - \overline{x}\right)^2.$$

- Une variance est toujours un nombre positif. L'écart-type est donc bien défini.
- Une série a une variance nulle si et seulement si toutes les valeurs de la série sont identiques.

Exemple.

- Avec Python, la variance et l'écart-type sont obtenus par les commandes `var()` et `std()` qui prennent en argument une liste.

Console

```
>>> np.var(L)
267615024984.00293
>>> np.std(L)
517315.20853731234
```

PROPOSITION — **Transformation affine**

Soient x une série statistique et $a, b \in \mathbb{R}$.
Si x' est la série statistique obtenue en modifiant chaque donnée x_i en $ax_i + b$, alors

$$V' = a^2 V \quad et \quad \sigma' = |a|\sigma$$

où V, σ, (resp. V', σ') désignent la variance et l'écart-type de x (resp. de x').

En pratique, on utilise la formule suivante pour calculer la variance.

THÉORÈME — **Formule de Koenig-Huygens**

Soit x une série statistique de modalités $(m_i)_{1\leqslant i\leqslant p}$, d'effectifs $(n_i)_{1\leqslant i\leqslant p}$ et de variance $\sigma(x)^2$, alors

$$\sigma(x)^2 = \left(\frac{1}{N}\sum_{i=1}^{N} {x_i}^2\right) - \overline{x}^2 = \left(\frac{1}{N}\sum_{i=1}^{p} n_i {m_i}^2\right) - \overline{x}^2.$$

Remarque. Si x^2 désigne la série dont chaque donnée est ${x_i}^2$. On démontre que $\frac{1}{N}\sum_{i=1}^{p} n_i {m_i}^2$ est la moyenne $\overline{(x^2)}$. La formule de formule de Koenig-Huygens devient alors

$$\sigma(x)^2 = \overline{(x^2)} - \left(\overline{x}\right)^2.$$

Exercice 34

✦✦ **L'écart type, un indicateur de dispersion par rapport à la moyenne**

Soit $x = (x_i)_{1\leqslant i\leqslant N}$ une série statistique avec $\overline{x}$ et $\sigma(x)$, respectivement la moyenne et l'écart-type de la série. Soit r le nombre d'éléments de la série statistique compris entre $\overline{x} - 2\sigma(x)$ et $\overline{x} + 2\sigma(x)$.

1. Montrer que
$$\sum_{k=1}^{N}\left(x_k - \overline{x}\right)^2 \geqslant 4\sigma(x)^2(N - r).$$
2. En déduire qu'au moins les trois quarts des éléments de la série statistique sont compris entre $\overline{x} - 2\sigma(x)$ et $\overline{x} + 2\sigma(x)$.

DÉFINITIONS — **Écart inter-quartile, étendue (hors-programme)**

Soit x une série statistique dont Q_1 et Q_3 représentent respectivement le premier quartile et troisième quartile.
- *Le nombre $Q_3 - Q_1$ s'appelle **l'écart inter-quartile**.*
- ***L'étendue** d'une série statistique est la différence entre la valeur la plus grande et la valeur la plus petite de cette série.*

Remarque. L'étendue est la longueur de l'intervalle $[Q_1, Q_3]$. Ce dernier contient des valeurs de X, réparties autour de la médiane, qui sont atteintes par la moitié des individus de l'échantillon. L'étendue et l'écart inter-quartile sont des indicateurs de dispersion. En particulier, ils ne varient pas si on augmente chaque valeur de la série par une même constante.

3 Représentations des séries statistiques

Diagramme des fréquences cumulées

Soit x une série de modalités m_1, m_2, m_p (rangés dans l'ordre croissant) et de fréquences $(f_i)_{i\in[\![1;p]\!]}$. Le diagramme des fréquences cumulées s'obtient en joignant par des segments, dans cet ordre, les points $\left(m_{i+1}, \sum_{j=1}^{i} f_j\right)$, pour $i \in [\![1,p]\!]$.

Voici un code pour tracer la courbe représentative de l'exemple 1 à l'aide de la commande `cumsum`.

Editeur

```
import matplotlib.pyplot as plt
import numpy as np
val=[8,10,12,14,15]
Eff=[1,2,4,1,1]
Freq=Eff/np.sum(Eff)
FC=np.cumsum(Freq)
plt.plot(val,FC)
plt.show()
```

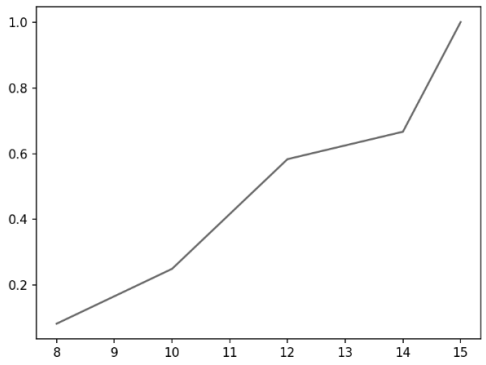


Diagramme en boîte

Considérons une série statistique dont on connaît :

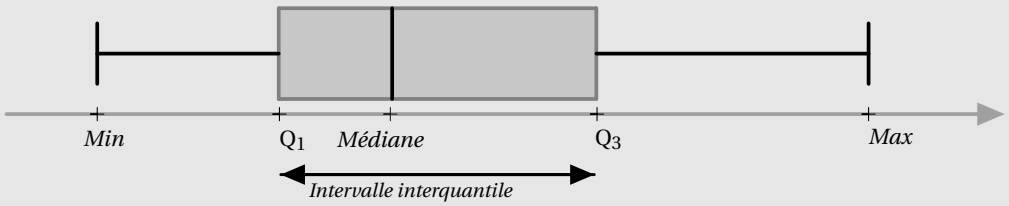
- Le premier quartile Q_1 et troisième quartile Q_3 : c'est-à-dire le plus petit nombre de la série tel qu'au moins 25% (resp. 75%) des données soient inférieures à ce nombre.
- La médiane et les valeurs extrêmes.

À partir de ces deux valeurs, on construit :

- L'intervalle interquartile $[Q_1;Q_3]$ contenant pratiquement 50% des valeurs de la série.
- L'écart interquartile $Q_3 - Q_1$.

DÉFINITION **Diagramme en boîte**

À partir d'une série statistique, on construit le diagramme en boîte par :



Vocabulaire. On parle plus communément de boîte à moustache.

Exemple. Ci-contre, le code pour tracer le diagramme en boîte de la liste L du nombre d'habitants par département.

Editeur

```
import matplotlib.pyplot as plt
import numpy as np

 plt.boxplot(L)
 plt.show()
```

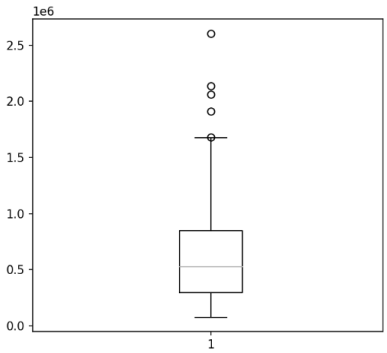


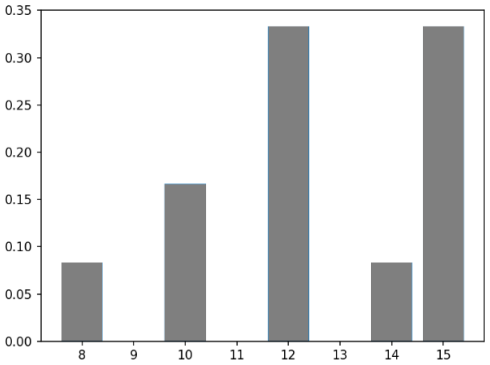
Diagramme en bâtons

Un diagramme en bâtons est un graphique où figure en abscisses les modalités $m_1, \cdots, m_p$ et en ordonnées les effectifs correspondants (pour un diagramme d'effectifs) ou les fréquences correspondantes (pour un diagramme de fréquences).

Un code possible afin d'afficher les données à l'aide de la fonction Python `bar`.

Editeur

```
val=[8,10,12,14,15]
Eff=[1,2,4,1,4]
Freq=Eff/np.sum(Eff)
plt.bar(val,Freq)
plt.show()
```

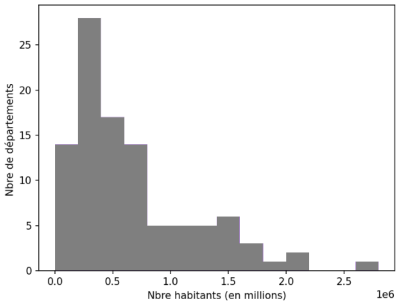


Histogrammes

Un histogramme donne une idée graphique de la répartition des valeurs du caractère sur l'échantillon. Ce graphique est obtenu en traçant, pour chaque $i \in [\![1, p]\!]$, le rectangle de base $[a_i, a_{i+1}]$ sur l'axe des abscisses et en ordonnées, l'effectif de la classe $[a_i, a_{i+1}]$.

Editeur

```
# Création d'un tableau avec les intervalles de
    longueurs 200 000 (habitants)
inter =np.linspace(0,2.8*10**6,15)
plt.hist(L, bins=inter)
plt.xlabel('Nbre habitants (en millions)')
plt.ylabel('Nbre de départements ')
plt.show()
```



4 Statistiques bivariées

Dans la suite, x, y désigne deux séries statistiques. La question est de savoir dans quelle mesure l'une des deux, dite **expliquée**, dépend de l'autre, dite **explicative**.

4.1 Premières définitions

DÉFINITION — **La covariance empirique**

*Soient $x = (x_i)_{1 \leqslant i \leqslant N}$ et $y = (y_i)_{1 \leqslant i \leqslant N}$, deux séries statistiques. On définit la **covariance** des deux séries par*

$$\mathrm{Cov}(x, y) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y}).$$

*Si les séries x et y ne sont pas constantes, on définit aussi le **coefficient de corrélation linéaire empirique** par*

$$\rho(x, y) = \frac{\mathrm{Cov}(x, y)}{\sigma(x)\sigma(y)}.$$

Remarques.

→ $\rho(x, y)$ est un réel de $[-1, 1]$.

→ On a l'égalité $\rho(x,y) = \pm 1$ si et seulement s'il existe a et b tels que pour tout indice i, $y_i = ax_i + b$. Dans ce cas, le signe de a est le même que celui de $\rho(x,y)$.

PROPOSITION — **Calcul de la covariance**

Soient $x = (x_i)_{1\leqslant i\leqslant N}$ et $y = (y_i)_{1\leqslant i\leqslant N}$, deux séries statistiques. On a

$$\mathrm{Cov}(x,y) = \overline{x*y} - \overline{x}\cdot\overline{y}$$

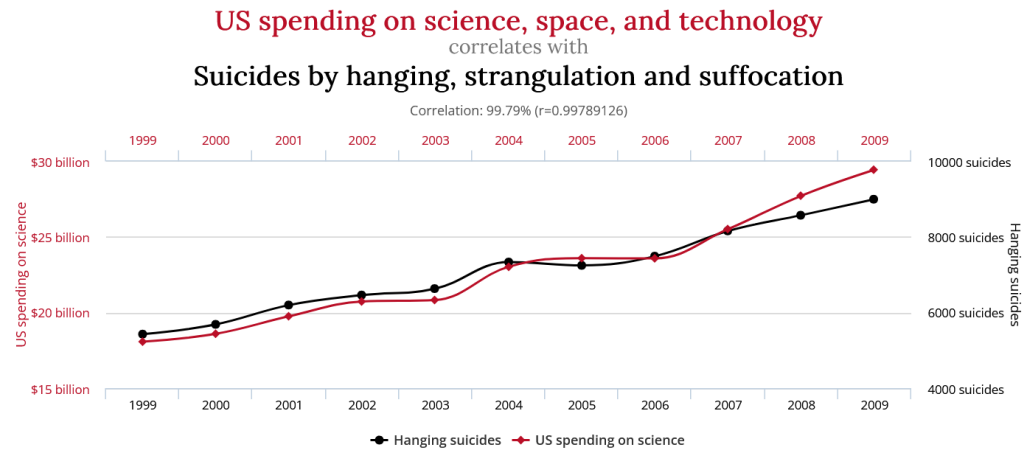
où $\overline{x*y}$ désigne la moyenne de la série $x*y = (x_i\cdot y_i)_{1\leqslant i\leqslant N}$.
Dès lors, la commande Python pour obtenir la covariance empirique est

```
np.mean(x*y)-np.mean(x)*np.mean(y)
```

et pour le coefficient de corrélation

```
(np.mean(x*y)-np.mean(x)*np.mean(y))/(np.std(x)*np.std(y))
```

Corrélation n'est pas causalité



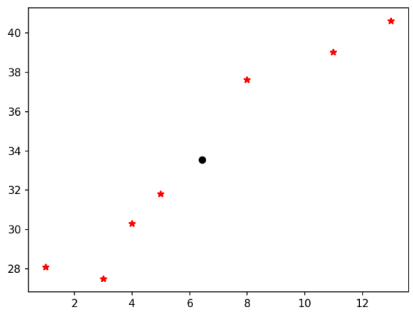
4.2 Nuage de points

DÉFINITION — **Nuage de points**

Soient $x = (x_i)_{1\leqslant i\leqslant N}$ et $y = (y_i)_{1\leqslant i\leqslant N}$, deux séries statistiques.
*Le **nuage de points** associé à ces deux séries est l'ensemble des points M_i du plan de coordonnées (x_i, y_i) où $i \in [\![1,N]\!]$.*

Remarque. Le point $(\overline{x},\overline{y})$ est le **point moyen** du nuage.

• Voici le code pour tracer un nuage de points associé à deux séries statistiques de même effectif, ainsi que son point moyen.



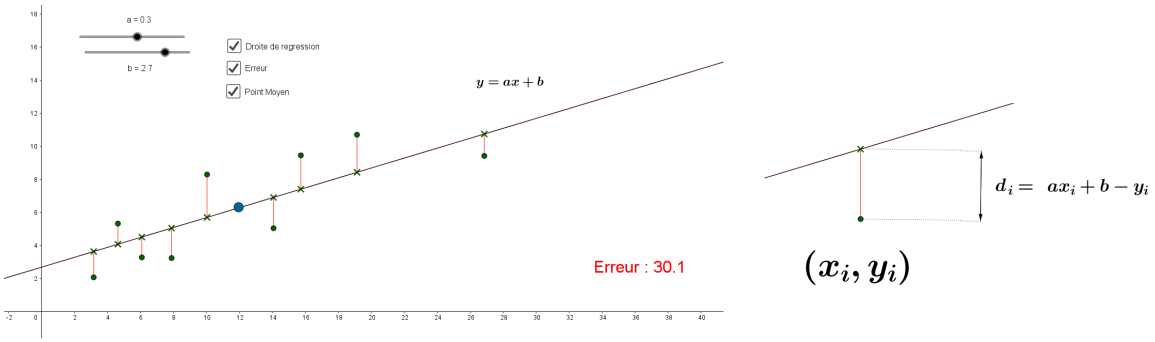
Editeur

```
import matplotlib.pyplot as plt
import numpy as np

# Exemples avec deux séries
x=np.array([1,3,4,5,8,11,13])
y=np.array([28.1,27.5,30.3,31.8,37.6,39,40.6])


# Tracé du nuage de points et du point moyen

plt.plot(x,y,'r*')
plt.plot(np.mean(x),np.mean(y),'ko')
```

4.3 Problème des moindres carrés et droite de régression

Considérons n points de $\mathbb{R}^2$, $(x_1, y_1), \ldots, (x_n, y_n)$ non alignés verticalement. On cherche la droite qui « approxime » au mieux ces n points. Si on note $y = ax + b$, l'équation d'une droite, on cherche à minimiser l'erreur

$$\mathrm{E}_r = \sum_{i=1}^{n} {d_i}^2 = \sum_{i=1}^{n} (ax_i + b - y_i)^2.$$



Point de vue algèbre linéaire

$$\mathrm{X} = \begin{bmatrix} a \\ b \end{bmatrix}, \quad \mathrm{A} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \quad \text{et} \quad \mathrm{B} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \text{de sorte que} \quad \mathrm{AX} - \mathrm{B} = \begin{bmatrix} ax_1 + b - y_1 \\ ax_2 + b - y_2 \\ \vdots \\ ax_n + b - y_n \end{bmatrix}.$$

Si on considère le produit scalaire canonique sur $\mathscr{M}_{n,1}(\mathbb{R})$ et la norme associée $\mathrm{E}_r = \|\mathrm{AX} - \mathrm{B}\|^2$. Les deux colonnes de la matrice A forment une famille libre (les points ne sont pas alignés verticalement). La matrice A est de rang 2. D'après le théorème précédent, il existe un seul vecteur minimisant $\|\mathrm{AX} - \mathrm{B}\|$. Calculons ce vecteur X_0 à partir des résultats sur les pseudo-solutions. On a

$${}^t\mathrm{AA} = \begin{bmatrix} \sum_{i=1}^{n} {x_i}^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & n \end{bmatrix} \in \mathscr{M}_2(\mathbb{R}).$$

${}^t\mathrm{AA}$ est inversible avec : $$\left({}^t\mathrm{AA}\right)^{-1} = \frac{1}{n\sum_{i=1}^{n} {x_i}^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \begin{bmatrix} n & -\sum_{i=1}^{n} x_i \\ -\sum_{i=1}^{n} x_i & \sum_{i=1}^{n} {x_i}^2 \end{bmatrix}.$$

Ceci permet d'expliciter les vecteur X, puis ses composantes a et b. Si on introduit les variances et covariances empiriques, on obtient

$$a = \rho(x, y)\sqrt{\frac{\mathbf{V}(y)}{\mathbf{V}(x)}} \quad \text{et} \quad b = \overline{y} - a\overline{x}.$$

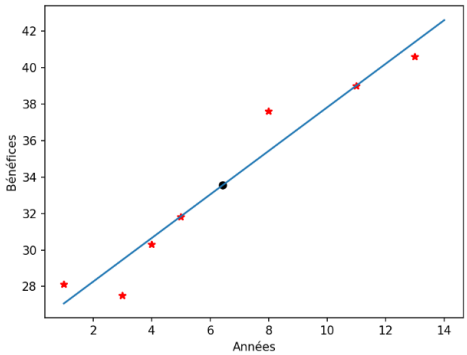
L'équation de la droite est alors : $$y - \overline{y} = \rho(x, y)\frac{\sigma(y)}{\sigma(x)}(x - \overline{x}).$$

Remarque. Avec cette dernière expression, on constate que le point moyen appartient à la droite de régression.

Exemple. On peut compléter l'exemple Python précédent en affichant la droite de régression. On comprend aussi sur cet exemple que la droite de régression peut être utilisée pour faire de la prédiction en prolongeant la droite. Précisons aussi qu'il existe des commandes Python plus directes pour afficher la droite de régression.

Editeur

```
# Tracé de la droite de régression

plt.xlabel('Années')
plt.ylabel('Bénéfices')
COVxy=np.mean(x*y)-np.mean(x)*np.mean(y)
a=COVxy/np.std(x)**2
b=np.mean(y)-a*np.mean(x)
t=np.linspace(1,14,2)
plt.plot(t,a*t+b)
```



Remarque. Un des inconvénients de la droite de régression est qu'elle est très sensible aux valeurs extrêmes : une seule valeur très éloignée de la droite modifie beaucoup le coefficient de corrélation linéaire et donc la pente de la droite. En particulier, la droite est très sensible aux erreurs de mesure.

Point de vue fonctions de plusieurs variables

Exercice 35

✦✦ *D'après HEC 2008 E*

Soient x, y deux séries statistiques avec $\sigma(x) \neq 0$. Soit f la fonction définie sur $\mathbb{R}^2$ par :

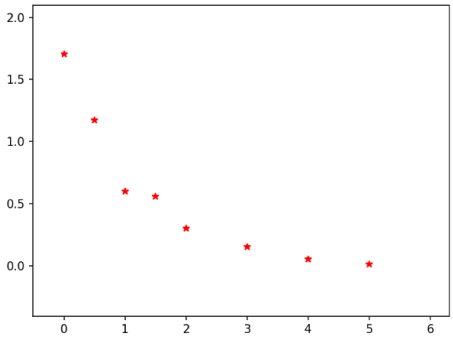
$$\forall (a,b) \in \mathbb{R}^2, \quad f(a,b) = \sum_{k=1}^{n} (ax_k + b - y_k)^2.$$

1. Justifier que f est de classe $\mathscr{C}^2$ sur $\mathbb{R}^2$.
2. **a)** Écrire le système d'équations $\mathscr{S}$ permettant de déterminer les points critiques de f.
   b) Résoudre le système $\mathscr{S}$.
   En déduire que f admet un unique point critique $(\hat{a}, \hat{b})$ que l'on exprimera en fonction de $\overline{x}$, $\overline{y}$, $\sigma(x)$ et $\text{cov}(x,y)$.
   c) Montrer que ce point critique correspond à un minimum local de f.
   d) Établir la formule suivante : $f(\hat{a}, \hat{b}) = n\sigma(y)^2\left(1 - \rho(x,y)^2\right)$.
3. **a)** Montrer que l'on a : $|\rho(x,y)| \leqslant 1$.
   b) Que peut-on dire du nuage de points lorsque $|\rho(x,y)| = 1$?

Remarque. Retenons que lorsque le coefficient de corrélation linéaire est proche de ± 1, la droite de régression approche bien le nuage de points. Dans ce cas, on peut modéliser la dépendance et faire des prédictions.
De plus, si le coefficient de corrélation est positif, les données x_i « auront tendance » à augmenter lorsque y_i augmente et inversement si le coefficients est négatif.

4.4 Un exemple d'ajustement linéaire

Il arrive parfois que le nuage ne s'accorde pas à une droite mais à une courbe d'une fonction classique.

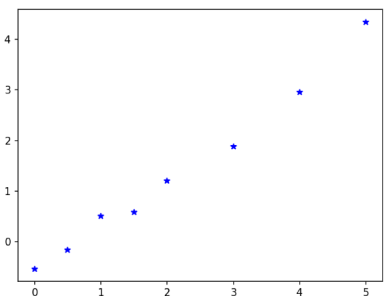


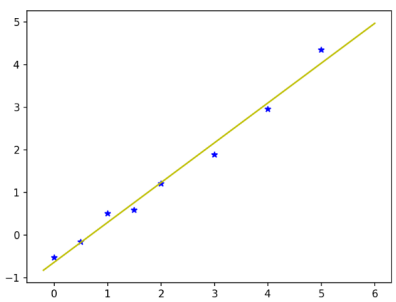
On peut alors supposer que le nuage s'organise autour de la courbe d'équation $y = f(ax+b)$ pour une certaine fonction f. Si f est bijective, on introduit la série $z = f^{-1}(x)$. Prenons l'exemple des séries :

Editeur

```
x=np.array([0,0.5,1,1.5,2,3,4,5])
y=np.array([1.707, 1.173, 0.601, 0.558,
    0.299,0.152, 0.052, 0.013])

plt.plot(x,y,'r*')
```

Dans l'exemple, on va tester avec la fonction f définie sur $\mathbb{R}$ par $f(t) = \exp(-t)$.
On procède alors à une régression linéaire sur les séries x, z pour estimer les réels a et b.





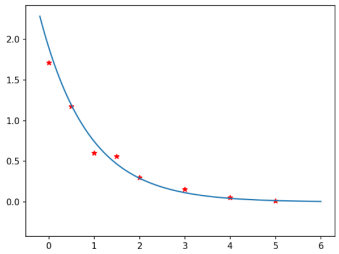
Console

```
>>> a
0.9349559793558717
>>> b
-0.6381866319796354
```

Testons le résultat, en comparant la courbe obtenue avec le nuage de points.

Editeur

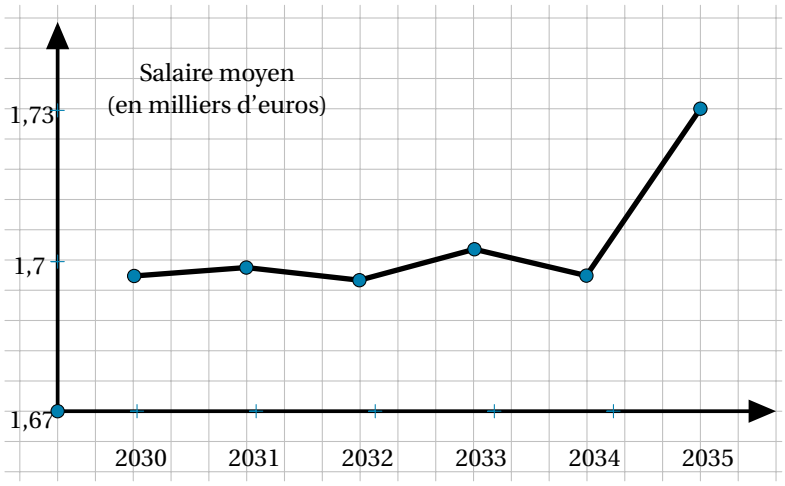
```
z=-np.log(y)
COVxz=np.mean(x*z)-np.mean(x)*np.mean(z)
a=COVxz/np.std(x)**2
b=np.mean(z)-a*np.mean(x)
plt.plot(x,y,'r*')
t=np.linspace(-0.2,6,100)
plt.plot(t,np.exp(-(a*t+b)))
plt.show()
```



Exercice 36. ✦✦ Quelques biais statistiques

Chercher l'erreur ou le biais dans les raisonnements suivants.

1. Dans la classe d'ECG, on compte qu'en moyenne les élèves ont 1.7 frères et sœurs. Ce qui donne 2.7 enfants par femme alors que le nombre d'enfants par femme n'est que de 1.8 dans la population française. Donc les enfants de familles nombreuses font plus facilement des études.
2. L'espérance de vie des professeurs de mathématiques est de 82 ans, soit 3 ans de plus que l'espérance de la population française. Donc les mathématiques sont bonnes pour la santé!
3. Lors de la seconde Guerre Mondiale, la Royal Air Force souhaite améliorer le taux de retour de ses bombardiers partis frapper les positions Allemandes. Les ingénieurs de la R.A.F décident d'étudier la localisation des impacts de balles sur les avions à leur retour de mission, puis de renforcer le blindage sur les zones les plus atteintes.
4. En 1936, le démocrate Franklin D. Roosevelt et le Républicain Alfred M. Landon concoururent pour la présidence des États-Unis. Juste avant l'élection, le journal *Literary Digest* réalisa un sondage de grande ampleur : 10 millions de bulletins de sondage sont distribués aux abonnés du magazine et à des gens figurant au bottin téléphonique. Après plus de 2 millions de réponses, Alfred Landon est annoncé président des États-Unis!
5. Comme le montre le graphe suivant, la nouvelle politique économique démarrée en 2034 a eu des effets considérables sur le salaire moyen des français.



6. Deux traitements contre les calculs rénaux présentent les résultats en fonction de la taille des calculs

petits calculs		gros calculs	
Traitement 1	Traitement 2	Traitement 1	Traitement 2
81 succès sur 87	234 succès sur 270	192 succès sur 263	55 succès sur 80

En regroupant les résultats, on obtient

Traitement 1	Traitement 2
78% (273 succès sur 350)	83% (289 succès sur 350)

Le second traitement est donc le plus efficace car il a le plus gros pourcentage de réussite.

7. Dans une université, à l'examen de la licence de biologie, les filles ont mieux réussi que les garçons et à l'examen de la licence de physique, les filles ont, là encore, mieux réussi que les garçons. Pourtant, en regroupant les résultats des deux licences, on découvre que les garçons ont mieux réussi que les filles. Il y a donc une erreur dans les résultats de la licence en faveur des garçons.

Les faits sont têtus, il est plus facile de s'arranger avec les statistiques.

MARK TWAIN