3
Intervalles de confiance

I
Intervalles de fluctuation

Q 1. *On considère une variable aléatoire X de carré intégrable, définie sur un espace probabilisé $(\Omega, \mathcal{A}, \mathsf{P})$. Cette variable aléatoire admet une espérance $m = \mathsf{E}(X)$ et un écart type*

$$\sigma = \sqrt{\mathsf{E}(X^2) - \big[\mathsf{E}(X)\big]^2}.$$

D'après l'inégalité de Bienaymé-Tchebychev,

$$\forall\, \varepsilon > 0, \qquad \mathsf{P}\big(|X - m| \geqslant \varepsilon\big) \leqslant \frac{\sigma^2}{\varepsilon^2}.$$

Q 1.a *Justifier l'existence de m et de σ.*
Q 1.b *Pourquoi l'inégalité de Bienaymé-Tchebychev n'a-t-elle d'intérêt que pour $\varepsilon > \sigma$?*
Q 1.c *Pour $\varepsilon > \sigma$, on considère l'événement*

$$A_\varepsilon = \big[X \in [m - \varepsilon, m + \varepsilon]\big] \in \mathcal{A}.$$

Que dire de la probabilité $\mathsf{P}(A_\varepsilon)$?

1. Si la variable aléatoire X suit une loi de probabilité usuelle (Bernoulli, binomiale, géométrique, Poisson...), le module `numpy.random` permet de simuler un échantillon de N réalisations de X. NB : au besoin, N peut être un tuple d'entiers.

```
import numpy.random as rd
```

1.1 Lois discrètes

Les commandes suivantes renvoient un échantillon de N réalisations d'une variable aléatoire discrète X suivant respectivement

- la loi uniforme sur $[\![a, b[\![$ (avec $a < b$ entiers) ;
- la loi de Bernoulli $\mathscr{B}(p)$ (avec $0 < p < 1$) ;
- la loi binomiale $\mathscr{B}(n, p)$ avec $n \in \mathbb{N}^*$ et $0 < p < 1$;
- la loi géométrique $\mathscr{G}(p)$ avec $0 < p < 1$;
- la loi de Poisson $\mathscr{P}(\lambda)$ avec $\lambda > 0$.

```
ech = rd.randint(a, b, N)  # loi uniforme discrète
ech = rd.binomial(1, p, N) # loi de Bernoulli
ech = rd.binomial(n, p, N) # loi binomiale
ech = rd.geometric(p, N)   # loi géométrique
ech = rd.poisson(lbd, N)   # loi de Poisson
```

1.2 Modèle d'urne

La fonction `choice` permet de simuler facilement un tirage au hasard de N éléments dans un ensemble de N_0 éléments, avec ou sans remise, avec ou sans hypothèse d'équiprobabilité.
Le mode d'emploi précis et quelques exemples se trouvent dans l'aide (exécuter `help(rd.choice)` dans le terminal).

1.3 Lois continues

Les commandes suivantes renvoient un échantillon de N réalisations d'une variable aléatoire continue X suivant respectivement

— la loi uniforme sur $[0,1[$;

— la loi exponentielle $\mathscr{E}(\lambda)$ avec paramètre $\lambda > 0$;

— la loi gamma de paramètres $n \in \mathbb{N}^*$ et $\lambda > 0$;

— la loi normale (ou gaussienne) $\mathscr{N}(m,s)$ d'espérance $m \in \mathbb{R}$ et d'écart type $s > 0$.

```
ech = rd.random(N)           # loi uniforme
ech = rd.exponential(lbd, N) # loi exponentielle
ech = rd.gamma(n, lbd, N)    # loi gamma
ech = rd.normal(m, s, N)     # loi normale
```

Q 2. *On donne la fonction suivante.*

```
def proportion(ech, m, s):
  a, b = m-3*s, m+3*s
  nb = 0
  for valeur in ech:
    nb += (a<valeur)*(valeur<b)
  return 100*nb/len(ech)
```

Pour différentes lois usuelles, appliquez cette fonction à un échantillon `ech` *de taille N assez grande (au moins quelques centaines), en prenant pour m et s, l'espérance et l'écart type de la loi choisie.*

Q 2.a *Que dire des valeurs renvoyées par cette fonction ?*

Q 2.b *Que calcule cette fonction ? Comment expliquer les valeurs observées ?*

II

Théorème limite fondamental

2. Loi normale centrée réduite

Lorsqu'on mesure une valeur, le résultat de la mesure est entaché d'erreurs et d'incertitudes, si bien que différentes mesures donnent des résultats différents.

La **loi normale** tire son nom du fait que, *normalement*, les résultats des différentes mesures se répartissent de part et d'autre de la valeur exacte suivant une courbe en cloche.

Plus précisément, une variable aléatoire $X : \Omega \to \mathbb{R}$ suit la loi normale centrée réduite si, et seulement si,

$$\forall\, a < b, \qquad \mathsf{P}(a \leqslant X \leqslant b) = \int_a^b \frac{1}{\sqrt{2\pi}}\, e^{-t^2/2}\, \mathrm{d}t.$$

En particulier,

$$\int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}}\, e^{-t^2/2}\, \mathrm{d}t = 1$$

et il est important de savoir que

$$\int_{-3}^{3} \frac{1}{\sqrt{2\pi}}\, e^{-t^2/2}\, \mathrm{d}t > 0,99.$$

On peut vérifier expérimentalement cette inégalité avec la fonction `proportion` définie plus haut **[Q2.]**.

3. Le Théorème limite fondamental (*central limit theorem* en anglais) assure que, sous des hypothèses fréquemment vérifiées, une variable aléatoire centrée et réduite suit approximativement la loi normale centrée réduite.

En particulier, si X suit la loi binomiale $\mathscr{B}(n,p)$ avec n assez grand, alors

$$\forall\, a < b, \qquad \mathsf{P}\Big(a \leqslant \frac{X - np}{\sqrt{np(1-p)}} \leqslant b\Big) \approx \int_a^b \frac{1}{\sqrt{2\pi}}\, e^{-t^2/2}\, \mathrm{d}t$$

ou si X suit la loi de Poisson $\mathscr{P}(\lambda)$ avec λ assez grand, alors

$$\forall\, a < b, \qquad \mathsf{P}\Big(a \leqslant \frac{X - \lambda}{\sqrt{\lambda}} \leqslant b\Big) \approx \int_a^b \frac{1}{\sqrt{2\pi}}\, e^{-t^2/2}\, \mathrm{d}t.$$

4. Le module `scipy.special` contient une fonction `erf` liée à la loi normale centrée réduite.

```
from scipy.special import erf
```

Par définition de la fonction `erf`,

$$\forall\, z \geqslant 0, \qquad \mathrm{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-u^2}\, \mathrm{d}u = \frac{1}{\sqrt{\pi}} \int_{-z}^{z} e^{-u^2}\, \mathrm{d}u$$

et par définition de la loi normale centrée réduite, si la variable aléatoire Z suit la loi $\mathscr{N}(0,1)$, alors

$$\forall\, z \geqslant 0, \qquad \mathsf{P}(|Z| \geqslant z) = 1 - \mathsf{P}(|Z| \leqslant z) = 1 - \int_{-z}^{z} e^{-t^2/2} \frac{\mathrm{d}t}{\sqrt{2\pi}}.$$

Le changement de variable $t = \sqrt{2}\, u$ permet d'en déduire que

$$\forall\, z \geqslant 0, \qquad \mathsf{P}(|Z| \geqslant z) = 1 - \mathrm{erf}(z/\sqrt{2}).$$

Q 3. *On se propose ici de vérifier la qualité de l'approximation en comparant un histogramme avec la densité de la loi normale centrée réduite.*
La commande `hist` *se trouve dans le module* `matplotlib.pyplot`.

```
import matplotlib.pyplot as plt
```

Q 3.a *Définir un échantillon* `ech` *de $N = 10^5$ valeurs prises par une variable aléatoire X de loi $\mathscr{B}(n, p)$ avec $n = 100$ et $p = 0,5$. En notant m et s l'espérance et l'écart type de X, on construit l'histogramme des valeurs centrées réduites avec la commande suivante.*

```
plt.hist((ech-m)/s, bins=20, density=True)
```

NB : On pourra librement modifier la valeur de `bins` *(= le nombre de sous-intervalles utilisés pour construire l'histogramme).*
Il reste à superposer le graphe de la densité normale centrée réduite.

```
x = np.linspace(-3, 3)
y = np.exp(-x**2/2)/np.sqrt(2*np.pi)
plt.plot(x, y, 'r')
```

La tradition indique que l'entier $n = 30$ est assez grand pour que l'approximation soit satisfaisante, pourvu que les produits np et $n(1-p)$ soient au moins égaux à 5. Commenter en faisant varier n et p.

Q 3.b *Recommencer avec une variable aléatoire suivant la loi de Poisson.*

III

Estimation d'une proportion

5. On cherche souvent à déterminer la proportion p d'une population qui présente une particularité :

- proportion d'objets manufacturés conformes au cahier des charges (contrôle de qualité) ;
- proportion de patients sur lesquels un médicament agit de manière bénéfique ou, inversement, provoque des effets indésirables (autorisation de mise sur le marché) ;
- proportion d'électeurs ayant l'intention de voter pour tel candidat à une élection (sondage d'opinion)...

6. Lorsqu'il n'est pas possible (trop cher, trop long ou simplement impossible) de procéder à un **recensement**, c'est-à-dire à étudier chaque individu de la population concernée, on procède à un **sondage** : on ne considère qu'une (petite) partie de la population concernée, dite **échantillon**, et on cherche à **estimer** la proportion p à partir de la proportion p_1 calculée sur l'échantillon étudié.

7. Deux problèmes se posent (au moins).
7.1 Dans quelle mesure la proportion mesurée p_1 permet-elle de calculer une valeur approchée de la proportion exacte p ?
On répondra en définissant les **intervalles de confiance** : à défaut de connaître la proportion exacte p, on saura comment l'encadrer à l'aide de la proportion mesurée p_1 et des caractéristiques du sondage.
7.2 En matière de sondage d'opinion, il est important de savoir si deux candidats recueillent à peu près la même proportion d'intentions de vote ou si l'un d'eux est vraiment avantagé.
En matière d'efficacité médicamenteuse, il est important de comparer l'effet d'un traitement à l'absence de traitement (ou d'un traitement placebo).
Plus généralement, en comparant les proportions p_1 et p_2 calculées sur deux échantillons, peut-on conclure qu'il existe une différence significative entre ces deux proportions ?
On répondra en comparant les intervalles de confiance : si les intervalles de confiance sont disjoints, c'est que les deux proportions mesurées sont **significativement différentes** ; si, au contraire, les intervalles de confiance se chevauchent, c'est que les deux proportions mesurées ne sont pas significativement différentes.

8. Modélisation
On note $\Theta =]0,1[$.
8.1 On considère un espace probabilisable $(\Omega, \mathcal{A})$ et une famille $(X_k)_{0\leqslant k<n}$ de variables aléatoires définies sur $(\Omega, \mathcal{A})$, ainsi qu'une famille $(\mathsf{P}_\theta)_{\theta\in\Theta}$ de mesures de probabilité sur $(\Omega, \mathcal{A})$ telles que, pour P_θ, les variables aléatoires $X_0, \ldots, X_{n-1}$ suivent la loi de Bernoulli $\mathscr{B}(\theta)$:

$$\forall\ \theta\in\Theta,\ \forall\ 0\leqslant k<n, \qquad \mathsf{P}_\theta(X_k=1)=\theta \quad \text{et} \quad \mathsf{P}_\theta(X_k=0)=1-\theta$$

et soient indépendantes :

$$\forall\ (\varepsilon_k)_{0\leqslant k<n}\in\{0;1\}^n, \quad \mathsf{P}_\theta(X_0=\varepsilon_0, X_1=\varepsilon_1, \ldots, X_{n-1}=\varepsilon_{n-1}) = \prod_{0\leqslant k<n} \mathsf{P}_\theta(X_k=\varepsilon_k).$$

8.2 L'entier n est le nombre d'individus qui constituent l'échantillon étudié.
La variable X_k représente la réponse de l'individu k : l'événement $[X_k=1]$ peut donc être interprété comme le fait que le k-ième individu envisage de voter pour tel candidat ou que le traitement médical a eu un effet bénéfique sur sa santé ou (dans ce cas, l'individu est un objet) qu'il est conforme au cahier des charges. On considère donc l'événement $[X_k=1]$ comme un *succès* lors de la k-ième mesure.
Sous P_θ, la loi de X_k est la loi de Bernoulli $\mathscr{B}(\theta)$, c'est-à-dire que le paramètre θ peut être considéré comme la **fréquence théorique** de succès.
8.3 La variable aléatoire

$$S_n = \sum_{0\leqslant k<n} X_k$$

est égale au nombre (aléatoire !) de variables égales à 1 parmi $X_0, \ldots, X_{n-1}$. La moyenne

$$M_n = \frac{S_n}{n}$$

est donc la *proportion de succès* lors des n mesures effectuées.
8.4 Le principe d'un sondage est de considérer que les mesures effectuées lors de ce sondage sont *une* valeur particulière

$$\big(X_0(\omega), X_1(\omega), \ldots, X_{n-1}(\omega)\big)\in\{0;1\}^n$$

du vecteur aléatoire $(X_k)_{0\leqslant k<n}$, valeur particulière qui permet de calculer la **fréquence empirique** de succès :

$$\theta_1 = \frac{M_n(\omega)}{n} = \frac{1}{n}\sum_{0\leqslant k<n} X_k(\omega).$$

8.5 Notre objectif est alors de donner une valeur approchée (c'est-à-dire un *encadrement*) de la fréquence théorique θ (inconnue) en nous fondant uniquement sur la donnée de la **taille** n de l'échantillon (connue) et de la fréquence empirique θ_1 (mesurée sur l'échantillon).
8.6 En répétant le sondage dans les mêmes conditions, la fréquence théorique θ serait toujours la même, contrairement aux fréquences empiriques $\theta_1, \theta_2, \ldots, \theta_N$ dont les valeurs fluctueraient au fil des sondages.

IV

Intervalles de confiance

9. On reprend le modèle présenté au [**8**].
9.1 Pour un réel $0<\alpha<1$ fixé (assez proche de 0 en pratique), on cherche des variables aléatoires

$$U_n=\varphi_n(M_n) \quad \text{et} \quad V_n=\psi_n(M_n)$$

telles que

$$\forall\ n\in\mathbb{N},\ \forall\ \theta\in\Theta, \quad \mathsf{P}_\theta(U_n\leqslant\theta\leqslant V_n)\geqslant 1-\alpha.$$

9.2 Comme les bornes U_n et V_n de cet intervalle sont des fonctions de M_n, on peut les calculer à partir des seules données de l'expérience, sans connaître la valeur du paramètre θ. Ces variables U_n et V_n sont appelées des **estimateurs de** θ.

9.3 En pratique, α est inférieur à 10%, donc $1-\alpha$ est assez proche de 100%. Il est donc *possible mais peu probable* que la valeur exacte de θ se trouve en dehors de l'intervalle calculé.
L'intervalle $[U_n, V_n]$ est donc appelé **intervalle de confiance** du paramètre θ au **niveau de confiance** $(1-\alpha)$.

10. Il importe de bien distinguer les concepts d'**intervalle de fluctuation** et d'**intervalle de confiance** :

— On demande à un intervalle de fluctuation de contenir l'essentiel des valeurs d'un échantillon

— tandis qu'on demande à un intervalle de confiance de contenir la valeur d'un paramètre inconnu.

10.1 La fonction `verif_IC` prend un échantillon de fréquences empiriques, calcule un intervalle de confiance (au moyen de la fonction `methode` passée en premier argument) pour chacune d'elles et renvoie la proportion d'intervalles de confiance qui contiennent effectivement le paramètre `th`.

```
def verif_IC(methode, th, n, alpha, Nb_simul):
  Ech = rd.binomial(n, th, Nb_simul)/n
  Nb_IC_corrects = 0
  for e in Ech:
    U, V = methode(e, n, alpha)
    Nb_IC_corrects += (U<th)*(th<V)
  return Nb_IC_corrects/Nb_simul
```

10.2 Étant données les bornes U et V d'un intervalle de confiance et un échantillon de fréquences, la fonction `verif_IF` calcule la proportion de fréquences qui appartiennent à l'intervalle de confiance. On évalue ainsi la qualité de l'intervalle de confiance considéré comme un intervalle de fluctuation.

```
def verif_IF(Ech, U, V):
  Nb_total = len(Ech)
  Nb_encadrements_corrects = np.sum((Ech>U)*(Ech<V))
  return Nb_encadrements_corrects/Nb_total
```

10.3 Nous allons voir différents méthodes de calcul d'intervalles de confiance.
La fonction suivante nous permettra de vérifier que les intervalles de confiance ainsi calculés méritent bien leur nom, mais qu'il serait parfois inapproprié de les considérer comme des intervalles de fluctuation.

```
def comparaison_IC_IF(th, methode, n, alpha, Nb_simul):
  # la fréquence empirique de référence
  f_empirique = rd.binomial(n, th)/n
  print("Taille de l'échantillon sondé : {: >6}".format(n))
  print("Fréquence empirique           : {:9.2f}%\n".format(100*f_empirique))
  # intervalle de confiance au niveau de confiance (1-alpha)
  U, V = methode(f_empirique, n, alpha)
  print("Intervalle de confiance pour p au niveau {}% : ".format(alpha))
  print("{:5.3f} < p < {:5.3f}\n".format(U, V))
  # vérification de la qualité de l'intervalle de confiance
  print("On effectue {} estimations de la fréquence théorique p.".format(Nb_simul))
  res = "{:6.2%}\n".format(verif_IC(methode, th, n, alpha, Nb_simul))
  print("Proportion d'intervalles de confiance contenant p : {}".format(res))
  print("On calcule {} fréquences.".format(Nb_simul))
  # un échantillon de fréquences empiriques (Nb_simul sondages)
  ech = rd.binomial(n, th, Nb_simul)/n
  res = "{:6.2%}".format(verif_IF(ech, U, V))
  print("Proportion de fréquences dans l'intervalle de confiance : {}".format(res))
```

IV.1 Avec l'inégalité de Bienaymé-Tchebychev

11. Sous P_θ, la loi de S_n est la loi binomiale $\mathscr{B}(n,\theta)$. Par conséquent, l'espérance de M_n est égale à θ et son écart type à $\sqrt{n\cdot\theta\cdot(1-\theta)}$.
D'après l'inégalité de Bienaymé-Tchebychev, pour tout $\varepsilon > 0$,

$$\forall\,\theta\in\Theta, \forall\, n\in\mathbb{N}^*,\quad \mathsf{P}_\theta\big(\big|M_n-\theta\big|\geqslant\varepsilon\big)\leqslant\frac{\theta(1-\theta)}{n\varepsilon^2}\leqslant\frac{14}{n\varepsilon^2}.$$

On en déduit que

$$\forall\,\theta\in\Theta,\ \forall\, n\in\mathbb{N}^*,\quad \mathsf{P}_\theta\big(M_n-\varepsilon\leqslant\theta\leqslant M_n+\varepsilon\big)\geqslant 1-\frac{1}{4n\varepsilon^2}.$$

12. Fixons $0<\alpha<1$ — sachant que plus α est proche de 0, plus le résultat final sera fiable.
Pour tout $n\geqslant 1$, on choisit $\varepsilon>0$ de telle sorte que

$$\alpha=\frac{1}{4n\varepsilon^2}\qquad\text{c'est-à-dire}\qquad\varepsilon=\frac{1}{2\sqrt{n\alpha}}.$$

D'après l'inégalité de Bienaymé-Tchebychev, le paramètre inconnu θ appartient à l'intervalle

$$\Big[M_n-\frac{1}{2\sqrt{n\alpha}}\,,\quad M_n+\frac{1}{2\sqrt{n\alpha}}\Big]$$

avec une probabilité supérieure à $(1-\alpha)$.

Q 4. *Pour une valeur fixée $\theta\in\,]0,1[$ de* `th` *et un entier $n\geqslant 1$ (assez grand), on calcule une valeur de la variable aléatoire M_n.*

```
th, n = 0.15, 100                  # librement modifiables
f_empirique = rd.binomial(n, th)/n
```

Écrire une fonction `IC_BT(f_empirique, n, alpha)` *qui renvoie les bornes d'un intervalle de confiance du paramètre θ au niveau de confiance $(1-\alpha)$.*
Faire varier l'argument `alpha` *(entre 1% et 20% par exemple). Qu'observe-t-on ?*

Q 5. *Appliquer plusieurs fois la fonction* `comparaison_IC_IF` *à la méthode* `IC_BT` *pour différentes valeurs de l'argument* `alpha` *entre 1 et 20. On pourra fixer* `th` *à 0.15, l'entier n à 1000 (comme dans le cas des sondages d'opinion) et l'entier* `Nb_simul` *à 1000.*
La fréquence empirique est-elle assez proche de la fréquence théorique ?
Que penser de la largeur de l'intervalle de confiance ?
Dans quelle mesure l'inégalité de Bienaymé-Tchebychev est-elle trop grossière pour calculer un intervalle de confiance ?

IV.2 Avec le théorème limite fondamental

13. Pour tout $\theta\in\Theta$, pour la mesure P_θ, la variable aléatoire S_n suit la loi binomiale $\mathscr{B}(n,\theta)$.
13.1 Lorsque l'entier n est assez grand, la variable aléatoire centrée réduite

$$\frac{S_n-n\theta}{\sqrt{n\theta(1-\theta)}}$$

suit donc approximativement la loi normale centrée réduite (Théorème limite fondamental **[3]**). Par conséquent, d'après **[4]**,

$$\forall\,\varepsilon>0,\qquad\mathsf{P}_\theta\left(\left|\frac{S_n-n\theta}{\sqrt{n\theta(1-\theta)}}\right|\leqslant\varepsilon\right)\approx\operatorname{erf}\big(\varepsilon/\sqrt{2}\big).$$

13.2 La fonction `erf` réalise une bijection strictement croissante de $[0, +\infty[$ sur $[0, 1[$. (On peut le constater en traçant son graphe.)
Par conséquent, pour tout $0 < \alpha < 1$, il existe un, et un seul, $\varepsilon > 0$ tel que

$$\operatorname{erf}(\varepsilon/\sqrt{2}) = 1 - \alpha \in]0, 1[$$

et donc tel que

$$\forall\, \varepsilon > 0, \qquad \mathbf{P}_\theta\left(\left|\frac{S_n - n\theta}{\sqrt{n\theta(1-\theta)}}\right| \leqslant \varepsilon\right) \approx 1 - \alpha.$$

13.3 Le module `scipy.optimize` contient la fonction `fsolve` qui permet de calculer une valeur approchée de ε en fonction de α.

```
from scipy.optimize import fsolve

def epsilon(alpha): # alpha donné en %
  def fn_aux(z):
    return (1-0.01*alpha) - erf(z/np.sqrt(2))
  r = fsolve(fn_aux, 0)
  return r[0]
```

On retrouve ainsi les valeurs classiques du seuil ε.

α	20%	10%	5%	1%
ε	1,28	1,64	1,96	2,58

En pratique, il est donc légitime de supposer que $\varepsilon > 1$.

14. Le réel $\varepsilon > 1$ étant déterminé, on voit que l'encadrement

$$\left|\frac{S_n - n\theta}{\sqrt{n\theta(1-\theta)}}\right| \leqslant \varepsilon$$

est équivalent à l'encadrement

$$f_\varepsilon(\theta) = \theta - \varepsilon\sqrt{\frac{\theta(1-\theta)}{n}} \leqslant \frac{S_n}{n} \leqslant \theta + \varepsilon\sqrt{\frac{\theta(1-\theta)}{n}} = g_\varepsilon(\theta).$$

14.1 On vérifie facilement que la fonction f_ε est négative entre $\theta = 0$ et $\theta = \frac{\varepsilon^2}{1+\varepsilon^2}$ et qu'elle réalise une bijection strictement croissante de $[\frac{\varepsilon^2}{1+\varepsilon^2}, 1]$ sur $[0, 1]$.
De même, la fonction g_ε réalise une bijection strictement croissante de $[0, \frac{1}{1+\varepsilon^2}]$ sur $[0, 1]$ et prend des valeurs supérieures à 1 entre $\theta = \frac{1}{1+\varepsilon^2}$ et $\theta = 1$.
Il existe donc un unique couple $(\theta_1, \theta_2) \in [0, 1] \times [0, 1]$ tel que

$$f_\varepsilon(\theta_1) = \frac{S_n}{n} = g_\varepsilon(\theta_2).$$

L'encadrement $\left|\frac{S_n - n\theta}{\sqrt{n\theta(1-\theta)}}\right| \leqslant \varepsilon$ devient alors, compte-tenu des monotonies de f_ε et g_ε,

$$\theta_2 \leqslant \theta \leqslant \theta_1.$$

14.2 On peut approcher les valeurs θ_1 et θ_2 par dichotomie. Le code suivant exploite les propriétés de monotonie et effectue 16 itérations. Comme la largeur de l'encadrement initial est égale à 1, l'erreur entre la valeur calculée et la valeur exacte est inférieure à 2^{-16} et donc inférieure à 10^{-4} : cette précision est largement suffisante pour nos besoins.

```
def dicho_simplifiee(fn, freq_emp):
  a, b = 0, 1
  for n in range(15):
    c = (a+b)/2
    if fn(c)>freq_emp:
      b = c
    else:
      a = c
  return (a+b)/2
```

Q 6. *En utilisant les fonctions précédentes, écrire une fonction* `IC_TLF(f_empirique, n, alpha)` *qui renvoie un intervalle de confiance de la fréquence théorique au niveau de confiance* $(1-\alpha)$ *à partir de la fréquence empirique et de la taille n de l'échantillon.*

Version simplifiée

15. On peut aussi résoudre littéralement l'encadrement $\left|\frac{S_n - n\theta}{\sqrt{n\theta(1-\theta)}}\right| \leqslant \varepsilon$ (il s'agit en fait de résoudre une équation du second degré).
Lorsque le quotient ε^2/n est petit (on rappelle que, en pratique, ε est compris entre 1 et 3), un développement limité simple montre que l'encadrement considéré est à peu près équivalent à l'encadrement

$$M_n - \varepsilon\sqrt{\frac{M_n(1-M_n)}{n}} \leqslant \theta \leqslant M_n + \varepsilon\sqrt{\frac{M_n(1-M_n)}{n}}$$

où $M_n = S_n/n$ est la fréquence empirique.

Q 7. *Déduire de cet encadrement une fonction* `IC_TLF_simplifie(f_empirique, n, alpha)` *qui renvoie un intervalle de confiance de la fréquence théorique au niveau de confiance* $(1-\alpha)$ *à partir de la fréquence empirique et de la taille n de l'échantillon.*

Q 8. *À l'aide de la fonction* `comparaison_IC_IF`*, comparer les intervalles de confiance calculés à l'aide du Théorème limite fondamental (version normale ou version simplifiée) aux intervalles de confiance calculées avec l'inégalité de Bienaymé-Tchebychev.*

V

Différence statistiquement significative

Q 9. *Quelque temps avant une élection, un sondage d'opinion est réalisé en interrogeant un échantillon représentatif de* $n = 1000$ *personnes.*
Dix-sept pour cent des personnes interrogées se sont prononcées en faveur du candidat A pendant que quatorze pour cent des personnes interrogées manifestaient leur confiance au candidat B.
Le candidat A peut-il se réjouir du résultat de ce sondage ?

Réponses aux questions

I Intervalles de fluctuation

R 1.a Comme X est supposée de carré intégrable, elle admet une espérance et une variance (et donc un écart type).

R 1.b Pour $0 < \varepsilon < \sigma$, on majore une probabilité par un réel supérieur à 1, ce qui n'a aucun intérêt.

R 1.c Par passage au complémentaire, d'après l'inégalité de Bienaymé-Tchebychev,

$$\mathsf{P}(A_\varepsilon) \geqslant 1 - \frac{\sigma^2}{\varepsilon^2}.$$

En particulier, pour $\varepsilon = 3\sigma$, on a $\mathsf{P}(A_\varepsilon) \geqslant 1 - \frac{1}{9} = \frac{8}{9} \approx 90\%$.

R 2.a

Les valeurs renvoyées sont très proches de 100%.

R 2.b En Python, les booléens sont considérés comme des entiers : `True` est identifié à 1 et `False` à 0. Par conséquent, le produit `(a<valeur)*(valeur<b)` est égal à 1 si, et seulement si, le réel `valeur` est compris entre a et b. La boucle `for` compte donc le nombre de valeurs qui sont comprises entre a et b dans le tableau `ech`.

La valeur renvoyée par la fonction `proportion` est donc la proportion de valeurs comprises entre $m - 3\sigma$ et $m + 3\sigma$ dans l'échantillon `ech`.

D'après **[Q1.c]**, la variable aléatoire X appartient à l'intervalle $[m - 3\sigma, m + 3\sigma]$ avec une probabilité à peu près supérieure à 90%.

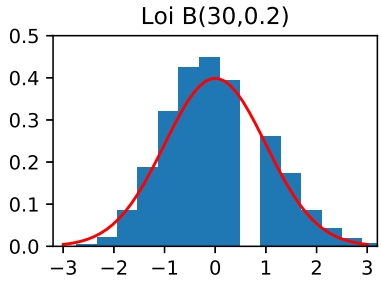
Les simulations faites montrent que l'inégalité de Bienaymé-Tchebychev est bien pessimiste : il semblerait que les valeurs de X appartiennent à l'intervalle $[m - 3\sigma, m + 3\sigma]$ avec une probabilité de l'ordre de 99% (au moins pour les lois usuelles considérées ici).

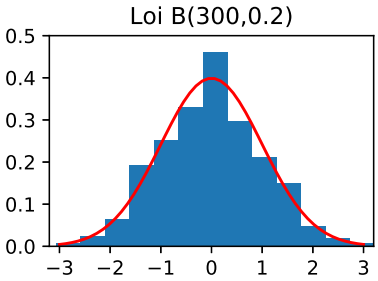
II Théorème limite fondamental

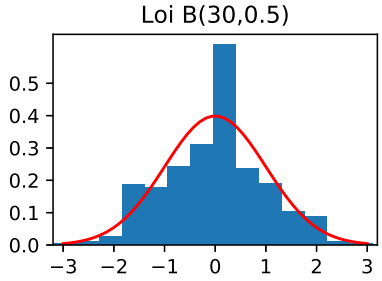
R 3.a Après quelques essais, on se lance dans l'écriture d'une fonction, ce qui permet de faire facilement varier les paramètres n et p.

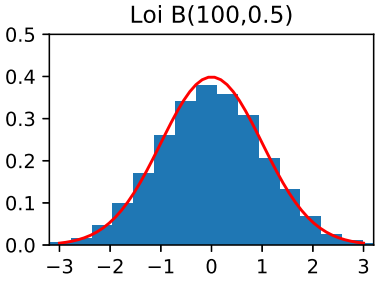
```
def TLF_bin(n, p):
  N = 10000
  m, s = n*p, np.sqrt(n*p*(1-p))
  ech = rd.binomial(n, p, N)
  plt.figure()
  plt.hist((ech-m)/s, bins=17, density=True)
  plt.plot(x, y, 'r')
  plt.xlim(-3.2, 3.2)
  plt.xticks([i for i in range(-3,4)])
  plt.yticks([0.1*i for i in range(6)])
  plt.title("Histogramme d'une variable de loi B({},{})".format(n, p))
```

Au fil des tentatives, on trouve des résultats plus ou moins satisfaisants...

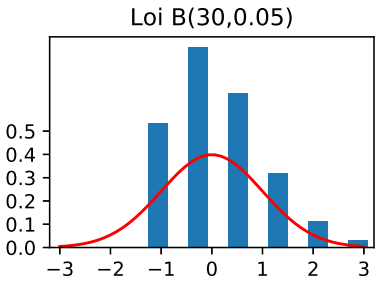


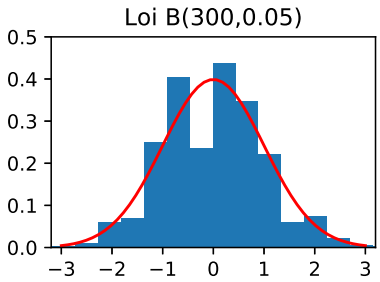


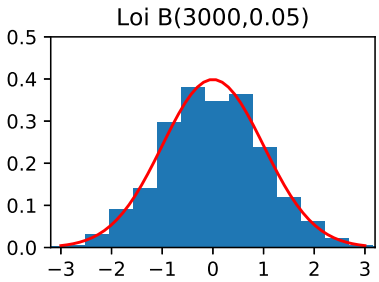




Manifestement, plus l'entier n est grand, meilleure est l'approximation !



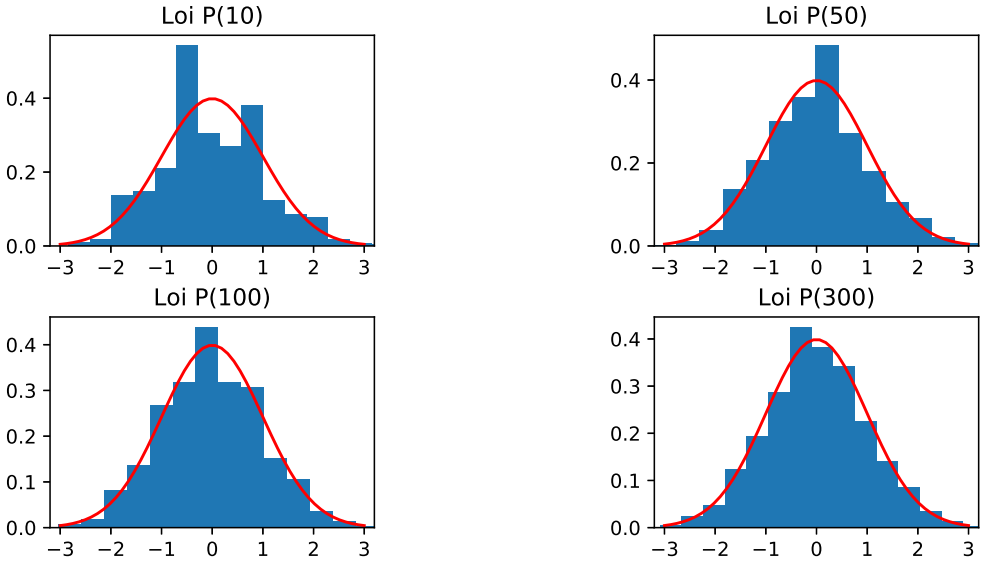




R 3.b Le code utilisé est analogue au code précédent.

```
def TLF_poi(lbd):
  N = 10000
  m, s = lbd, np.sqrt(lbd)
  ech = rd.poisson(lbd, N)
  plt.figure(figsize=(3,2))
  plt.hist((ech-m)/s, bins=17, density=True)
  plt.plot(x, y, 'r')
  plt.xlim(-3.2, 3.2)
  plt.xticks([i for i in range(-3,4)])
  plt.title("Histogramme d'une variable de loi P({})".format(lbd))
```

On constate de manière analogue que, plus le paramètre λ est grand, meilleure est l'approximation.



IV Intervalles de confiance

R 4.

On applique les formules trouvées ci-dessus. On peut trouver pratique d'exprimer le niveau de confiance en pourcentage, ce qui revient à substituer 100α à α.

```
def IC_BT(valeur, n, alpha):      # alpha exprimé en %
  eps = 5/np.sqrt(2*n*alpha)
  return (valeur-eps, valeur+eps)
```

Plus la valeur de α est proche de 0, plus l'intervalle de confiance est large (c'est-à-dire imprécis). Pour $\alpha = 1\%$, cet intervalle peut même contenir des valeurs négatives alors qu'on estime une probabilité...
Au contraire, plus α grandit, plus l'intervalle de confiance est précis. Mais dans ce cas, on constate parfois que la valeur exacte du paramètre se trouve en dehors de l'intervalle de confiance !
Il s'agit donc d'une affaire de *compromis* entre la **précision** (= intervalle de confiance étroit) et la **confiance** (= probabilité pour que l'intervalle calculé contienne bien la valeur estimée) : on ne peut pas gagner sur les deux tableaux.

R 5. Pour $\alpha = 5\%$, les différentes valeurs de la fréquence empirique semblent assez souvent proches de la fréquence théorique (entre 13% et 17% pour $\theta = 15\%$).
L'intervalle de confiance au niveau de confiance 95% semble au contraire démesurément large, il s'étend assez souvent de 8% à 22% !
Si on accepte un peu plus de risque, l'intervalle reste encore très large : l'intervalle de confiance au niveau de confiance 90% s'étend régulièrement de 11% à 18%.
On a l'impression de perdre sur les deux tableaux : l'intervalle de confiance reste assez large même avec un niveau de risque α assez élevé. On en déduit que l'inégalité de Bienaymé-Tchebychev est trop vague pour donner un intervalle de confiance raisonnable. Cette impression est confirmée par le fait que, pour la plupart des simulations, la quasi-totalité des valeurs de l'échantillon appartient à l'intervalle de confiance.

R 6. Si on a bien suivi la démarche précédente, le code est sans mystère : on calcule ε en fonction de α, on définit les deux fonctions auxiliaires f_ε et g_ε et on détermine les valeurs de θ_1 et θ_2 par dichotomie.

```
def IC_TLF(f_empirique, n, alpha):
  eps = epsilon(alpha)
  def f(th):
    return th-eps*np.sqrt(th*(1-th)/n)
  th1 = dicho_simplifiee(f, f_empirique)
  def g(th):
    return th+eps*np.sqrt(th*(1-th)/n)
  th2 = dicho_simplifiee(g, f_empirique)
  return th2, th1
```

R 7. On calcule ε en fonction de α, puis la demi-longueur de l'intervalle de confiance.

```
def IC_TLF_simplifie(f_empirique, n, alpha):
  eps = epsilon(alpha)
  delta = eps*np.sqrt(f_empirique*(1-f_empirique)/n)
  return f_empirique-delta, f_empirique+delta
```

R 8. Pour un même niveau de confiance $1 - \alpha$, les intervalles de confiance calculés avec le Théorème limite fondamental sont sensiblement plus étroits que ceux calculés avec l'inégalité de Bienaymé-Tchebychev : on a gagné en précision sans perdre en sûreté.
Les intervalles sont devenus si étroits qu'il est assez souvent impossible de les considérer comme des intervalles de fluctuation (ils contiennent assez souvent moins de la moitié des valeurs de l'échantillon).

V Différence statistiquement significative

R 9. Le sondage a publié la proportion de personnes favorables à chaque candidat *parmi l'échantillon des personnes interrogées*.
En admettant que l'échantillon choisi soit représentatif, on peut considérer que ces proportions sont des estimations fiables des proportions p_A et p_B d'électeurs favorables aux candidats A et B *dans l'ensemble du corps électoral*.
Nous pouvons calculer un intervalle de confiance pour les proportions p_A et p_B pour différents niveaux de risque.

```
f1, f2 = 0.17, 0.14
n = 1000
for alpha in [ 1, 5, 10, 20]:
  print("risque : {: >2}%".format(alpha))
  U1, V1 = IC_TLF(f1, n, alpha)
  print("IC1 = [ {:5.3f}% , {:5.3f}% ])".format(100*U1, 100*V1))
  U2, V2 = IC_TLF(f2, n, alpha)
  print("IC2 = [ {:5.3f}% , {:5.3f}% ])".format(100*U2, 100*V2))
  s = ""
  if V2>U1:
    s = "non "
  print("Différence {}significative".format(s))
```

Pour un risque α inférieur à 10%, les intervalles de confiance se chevauchent. Dans ce cas, il paraît audacieux d'affirmer que les proportions p_A et p_B sont nettement différentes.
Pour un risque α égal à 20%, les intervalles de confiance sont disjoints :

$$p_A \in [15,53\%,\ 18,58\%] \qquad p_B \in [12,65\%,\ 15,46\%].$$

Mais d'une part les intervalles de confiance sont quand même assez proches et d'autre part, on a une chance sur cinq de se tromper...
Le candidat A a donc l'air avantagé par rapport au candidat B mais il ne devrait quand même pas faire le malin.