Lycée Pierre-Gilles de Gennes 2024-2025

TP7 Python

Tests statistiques

1 Introduction : principe d'un test statistique

On considère une population Ω dont les individus possèdent un caractère X (modélisé par une variable aléatoire) dont la loi (ou un certains nombres de paramètres) sont inconnus.

Exemple : on dispose d'une pièce qu'on peut modéliser par une variable aléatoire suivant une loi de Bernoulli de paramètre inconnue p (la probabilité de faire pile par exemple).

On souhaite formuler une hypothèse sur le(s) paramètre(s) inconnu(s) (par exemple, une hypothèse sur sa valeur) et porter un jugement sur cette hypothèse (est-elle raisonnable ?), en se basant sur l'observation d'un échantillon prélevé.

Exemple (suite) : on souhaite déterminer si notre pièce est équilibrée ou non c'est-à-dire si $p = \frac{1}{2}$.
Pour cela, on peut par exemple lancer la pièce un grand nombre de fois et observer la proportion de piles et de faces obtenus.

Principe des tests :

- **Un test d'hypothèse** est une démarche qui a pour but de fournir une règle de décision permettant, sur la base de résultats d'échantillon, de faire un choix entre deux hypothèses.
- **Hypothèse nulle,** H_0 : c'est l'hypothèse que l'on cherche à tester. Par exemple, on fait l'hypothèse que la probabilité de faire pile vaut $\frac{1}{2}$.
- **Seuil de signification d'un test d'hypothèse** : c'est le risque α consenti à l'avance de rejeter à tort l'hypothèse H_0, alors qu'elle est vraie.
  On utilisera en général $\alpha = 5\%$.
- **Statistique de test ou variable d'échantillonnage** : c'est une variable aléatoire T (en lien avec le problème initial) dont on connait la loi sous l'hypothèse H_0.
  On sépare alors $\mathbb{R}$ en deux zones : une zone de rejet notée R_{rejet} et une zone de non-rejet notée $R_{\text{non-rejet}}$ telles que :

  $$\mathbb{P}(T \in R_{\text{rejet}}) = \alpha \quad ; \quad \mathbb{P}(T \in R_{\text{non-rejet}}) = 1 - \alpha.$$

  -**Utilisation du test** : à partir d'un échantillon, on calcule une valeur observée de T , notée t_{obs}.
  - Si $t_{obs} \notin R_{\text{non-rejet}}$ c'est-à-dire $t_{obs} \in R_{\text{rejet}}$ alors l'hypothèse H_0 est rejetée.
  - Sinon, l'hypothèse H_0 n'est pas rejetée.

Exemple (suite) : on reprend l'exemple précédent. On lance n fois la pièce et on note X_i la variable valant 1 si on obtient pile et 0 sinon.
La variable $\overline{X}_n = \dfrac{X_1 + \cdots + X_n}{n}$ donne la proportion de piles parmi l'échantillon de lancers et sous l'hypothèse $H_0 : p = \dfrac{1}{2}$, $n\overline{X}_n$ suit une loi $\mathcal{B}\left(n, \dfrac{1}{2}\right)$.

Travail demandé

1. Sous l'hypothèse $H_0 : p = \dfrac{1}{2}$, on sait que $n\overline{X}_n$ suit une loi $\mathcal{B}\left(n, \dfrac{1}{2}\right)$ donc
$$\mathbb{E}(n\overline{X}_n) = \frac{n}{2} \quad \text{et} \quad \mathbb{V}(n\overline{X}_n) = \frac{n}{4}.$$
En particulier, on a :
$$\mathbb{E}(\overline{X}_n) = \frac{1}{2} \quad \text{et} \quad \mathbb{V}(\overline{X}_n) = \frac{1}{4n}.$$
L'inégalité de Bienaymée-Tchebychev appliquée à $\overline{X}_n$ donne alors, pour tout $\varepsilon > 0$:
$$\mathbb{P}\left(\left|\overline{X}_n - \frac{1}{2}\right| \geq \varepsilon\right) \leq \frac{1}{4n\varepsilon^2}.$$
En passant au complémentaire, on obtient :
$$\mathbb{P}\left(\left|\overline{X}_n - \frac{1}{2}\right| \leq \varepsilon\right) \geq \mathbb{P}\left(\left|\overline{X}_n - \frac{1}{2}\right| < \varepsilon\right) \geq 1 - \frac{1}{4n\varepsilon^2}.$$
Enfin, il suffit de remarquer que $\mathbb{P}\left(\left|\overline{X}_n - \dfrac{1}{2}\right| \leq \varepsilon\right) = \mathbb{P}(\overline{X}_n \in \left[\dfrac{1}{2} - \varepsilon, \dfrac{1}{2} + \varepsilon\right])$ pour conclure.
2. En prenant comme statistique de test $\overline{X}_n$, la question précédente signifie exactement que l'intervalle $\left[\dfrac{1}{2} - \varepsilon, \dfrac{1}{2} + \varepsilon\right]$ est un intervalle de non-rejet avec un risque consenti inférieur à $\alpha = \dfrac{1}{4n\varepsilon^2}$.
3. On fixe $\varepsilon = 0.01$. On cherche la valeur de n à partir de laquelle
$$\frac{1}{4n\varepsilon^2} \leq 0.05.$$
On obtient $n = \dfrac{1}{4 \cdots 0.05 \cdots 0.01^2} = 50000$.
La longueur de la zone de non-rejet (ici 2ε) est liée à ce qu'on appelle la puissance du test : plus la zone de non-rejet est grande plus le risque d'accepter l'hypothèse H_0 alors qu'elle est fausse (risque de deuxième espèce) est grande.
On remarque donc ici que l'inégalité de Bienaymée-Tchebychev n'est pas très utilisable en pratique car il faut un échantillon très grands (50 000) pour un risque de deuxième espèce somme toute assez élevé (ici $2\varepsilon = 2\%$!).

2 Le théorème central limite

La zone de rejet obtenu dans le paragraphe précédent avec l'inégalité de Bienaymée-Tchebychev est assez mauvaise car pour avoir une précision acceptable (ε petit) et un seuil α de 5% il faut faire un (trop) grand nombre de tirages (si on pense à des sondages, on ne peut pas se permettre d'interroger 50 000 personnes).
Le but de cette partie est de visualiser expérimentalement ce qu'on appelle le théorème central limite (qu'on verra dans un chapitre ultérieur) puis de voir comment il permet d'améliorer le test précédent.

2.1 Approche expérimentale

Soit $(X_n)_{n\in\mathbb{N}}$ une suite de variables aléatoires indépendantes et de même loi dont on note μ l'espérance et σ l'écart-type, que l'on suppose non nul. On note pour tout $n \in \mathbb{N}^*$:

$$S_n = \frac{\sum_{k=1}^n X_k - n\mu}{\sigma\sqrt{n}} = \frac{\overline{X}_n - \mu}{\sqrt{\frac{\sigma^2}{n}}}.$$

On note f une densité d'une variable de loi $\mathcal{N}(0,1)$ et Φ sa fonction de répartition (voir l'annexe).

Travail demandé

1. Écrire une fonction qui renvoie $f(x)$.

```
def densite_normal(x):
    return 1/(np.sqrt(2*np.pi))*np.exp(-x**2/2)
```

2. Écrire une fonction `S_poisson(n,lambda_)` qui simule une réalisation de S_n lorsque les variables aléatoires $(X_n)_n$ suivent la loi $\mathcal{P}(\lambda)$.

```
def S_poisson(n,lambda_):
     temp = rd.poisson(lambda_,n)
     return (sum(temp)-n*lambda_)/np.sqrt(n*lambda_)
```

3. Simuler 10 000 réalisations de S_{1000} (pour différentes valeurs de `lambda_`) et afficher l'histogramme des valeurs obtenues.

```
N = 10000
lambda_ = 1
n = 1000
Poisson = []
for _ in range(N):
    Poisson.append(S_poisson(n,lambda_))

plt . hist (Poisson , density = True)

t = np.linspace(-5,5,1000)
densite = densite_normal(t)
plt.plot(t,densite)
```

On obtient le graphique 3 suivant :

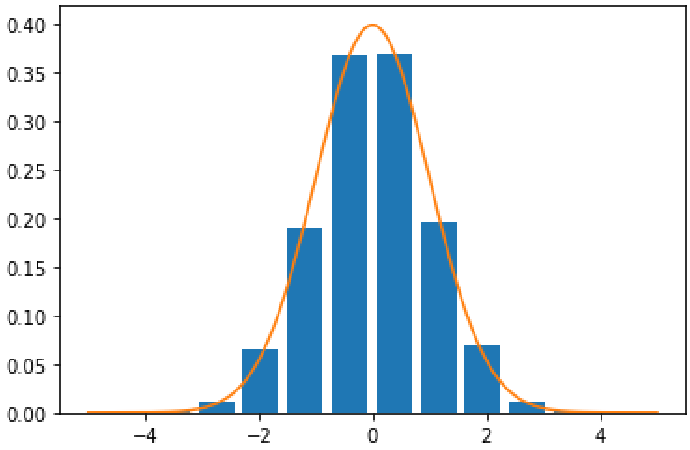

FIGURE 1 – Histogramme pour une loi de Poisson

4. En modifiant, le programme précédente on obtient les graphiques suivants :

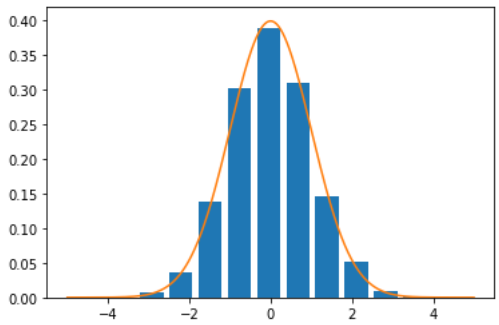

FIGURE 2 – Histogramme pour une loi de géométrique

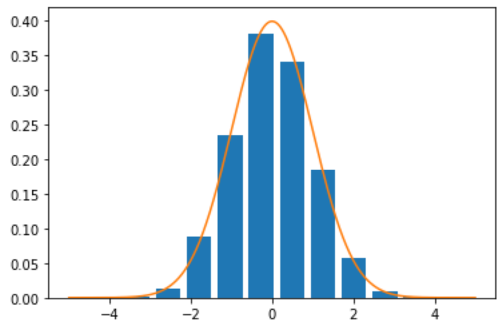

FIGURE 3 – Histogramme pour une loi exponentielle

5. On constate que, peu importe la loi, l'histogramme semble « coller ». Cela illustre le théorème central limite que nous verrons lors du dernier chapitre de mathématiques.

2.2 Le théorème central limite

Le phénomène observé sur les exemples précédents peut se traduire de la façon suivante :

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

Cela signifie que la suite $(S_n)_n$ « tend [1] » vers une variable aléatoire suivant une loi normale centrée réduite.

Pour n assez grand (en pratique $n > 30$), on pourra approximer $\mathbb{P}(a \leq S_n \leq b)$ par $\Phi(b) - \Phi(a)$.

1. Le sens précis derrière cette notion sera vu dans un chapitre ultérieur.

Exemple : on reprend le même exemple. Remarquons que pour tout n on a :

$$S_n = \sqrt{n}\frac{\overline{X}_n - p}{\sqrt{p(1-p)}}.$$

On en déduit que sous l'hypothèse $H_0 : p = \dfrac{1}{2}$:

$$\begin{aligned}\mathbb{P}\left(\frac{1}{2} - \frac{t}{2\sqrt{n}} \leq \overline{X}_n \leq \frac{1}{2} + \frac{t}{2\sqrt{n}}\right) &= \mathbb{P}(-t \leq S_n \leq t) \\ &\xrightarrow[n\to+\infty]{} 2\Phi(t) - 1.\end{aligned}$$

En prenant $t = t_\alpha$ tel que $\Phi(t_\alpha) = 1 - \dfrac{\alpha}{2}$ on obtient :

$$\begin{aligned}\mathbb{P}\left(\frac{1}{2} - \frac{t_\alpha}{2\sqrt{n}} \leq \overline{X}_n \leq \frac{1}{2} + \frac{t_\alpha}{2\sqrt{n}}\right) &= \mathbb{P}(-t_\alpha \leq S_n \leq t_\alpha) \\ &\simeq 1 - \alpha.\end{aligned}$$

L'intervalle $\left[p_0 - \dfrac{t_\alpha}{2\sqrt{n}}, p_0 + \dfrac{t_\alpha}{2\sqrt{n}}\right]$ est donc une zone de non rejet avec un risque consenti (environ) égal à α.

Travail demandé

1. Pour $\alpha = 5\%$, on peut déterminer que $t_\alpha \simeq 1.96$. On cherche n tel que :

$$p_0 + \frac{t_\alpha}{2\sqrt{n}} - (p_0 - \frac{t_\alpha}{2\sqrt{n}}) = 0.02$$

   c'est-à-dire :

$$\frac{t_\alpha}{\sqrt{n}} = 0.02.$$

   On trouve alors

$$n = \left(\frac{t_\alpha}{0.02}\right)^2 = 9604 < 50\ 000.$$

2. (a) Recopier le code suivant qui permet de lire et stocker la probabilité p dans la variable `p`.

```
fichier = open('../piece.txt')
p = int(fichier.read())/100
fichier.close()
```

   Le fichier contient la probabilité p (en %) que la pièce donne Pile.

   (b) Si on simule $n = 9604$ lancers de la pièce et qu'on note la fréquence d'apparition du Pile, on obtient une réalisation de la variable $\overline{X}_n$.

   D'après l'exemple, sous l'hypothèse $H_0 : p = \dfrac{1}{2}$, l'intervalle

   $\left[p_0 - \dfrac{1.96}{2\sqrt{n}}, p_0 + \dfrac{1.96}{2\sqrt{n}}\right]$ est donc une zone de non rejet avec un risque consenti (environ) égal à 5%.

```python
def piece_truquee():
    S = 0
    for k in range(9604):
        if rd.rand()<p:
            S += 1/9604
    if np.abs(S-1/2)<= 2*1.96/np.sqrt(9604):
        return "non rejet",S
    else:
        return "rejet",S
```

(c) Modifier la fonction précédente pour qu'elle affiche également une valeur approchée de p.

Cela revient à s'intéresser au problème au inverse : quelles sont les valeurs de p_0 pour lesquelles sous l'hypothèse $p = p_0$, la statistique de test tombe bien dans la zone de non rejet :

$$\overline{X}_n \in \left[p_0 - \frac{1.96}{2\sqrt{n}}, p_0 + \frac{1.96}{2\sqrt{n}}\right] \Longleftrightarrow p_0 \in \left[\overline{X}_n - \frac{1.96}{2\sqrt{n}}, \overline{X}_n + \frac{1.96}{2\sqrt{n}}\right].$$

Il y a donc 95% de chance que la vrai valeur de p tombe dans l'intervalle $\left[\overline{X}_n - \frac{1.96}{2\sqrt{n}}, \overline{X}_n + \frac{1.96}{2\sqrt{n}}\right]$ et $\overline{X}_n$ constitue donc une valeur approchée fiable à 95% de p.

Ce programme précédent renvoie dans la variable `S` cette valeur approchée.

En modifiant légèrement la fonction précédente, on peut avec la fonction `test` suivante, compter le nombre de fois où p tombe en effet dans l'intervalle ci-dessus. Quelques essais montrent que c'est le cas dans plus de 95% des cas.

```python
def piece_truquee2():
    S = 0
    for _ in range(9604):
        if rd.rand()<p:
            S += 1/9604
    if np.abs(S-1/2)<= 2*1.96/np.sqrt(9604):
        return 1
    else:
        return 0

def test(n=1000):
    s = 0
    for k in range(n):
        s += piece_truquee2()
    return s/n
```

3 Test de conformité à la moyenne sous hypothèse gaussienne

Les méthodes précédentes (inégalités de Markov et théorème central limite) comportent des inconvénients importants :

— la taille des échantillons nécessaires doit être importante ;

— la variance était connue (en fonction du paramètre à estimer).

On va donc chercher à mettre en place un test qui fonctionne sur les petits échantillons même lorsque la variance est inconnue. Pour simplifier, on se place dans le cas où le caractère à étudier X suit une loi normale $\mathcal{N}(\mu, \sigma^2)$ de paramètres μ et σ inconnus.

Contexte : on dispose d'une série statistique $x = (x_k)_{k \in [\![1,n]\!]}$ que l'on suppose être une réalisation d'un n-échantillon $(X_1, \ldots, X_n)$ d'une certaine variable aléatoire réelle X de loi $\mathcal{N}(\mu, \sigma^2)$.
Un nombre réel m étant donné, on souhaite tester la conformité de la série x à l'hypothèse :

$$H_0 \ : \mu = m.$$

Si on reprend la démarche du paragraphe 2, il est tentant de considérer les variables :

$$S_n = \frac{\overline{X}_n - \mu}{\sqrt{\frac{\sigma^2}{n}}}$$

pour construire une zone de rejet.
Cependant, le paramètre σ est ici inconnu donc on ne peut pas calculer la réalisation de S_n.

3.1 Moyenne et variance empiriques

Pour contourner ce problème, on va d'abord estimer la valeur de σ grâce à notre échantillon $(X_1, \ldots, X_n)$. On note pour tout $n \in \mathbb{N}^*$:

$$\overline{X}_n = \frac{1}{n} \sum_{k=1}^{n} X_k \text{ (moyenne empirique) } ; \ \Sigma_n^2 = \frac{1}{n-1} \sum_{k=1}^{n} (X_k - \overline{X}_n)^2 \text{ (variance empirique).}$$

Travail demandé

```python
def moyenne(L):
    return sum(L)/len(L)

def variance(L):
    s = 0
    moy = moyenne(L)
    for k in L:
        s +=(k-moy)**2
    return s/(len(L)-1)
```

On constate que $\overline{X}_n$ semble tendre vers $\mathbb{E}(X)$ et Σ_n^2 semble tendre vers $\mathbb{V}(X)$.

3.2 Loi du χ^2, loi de Student

Les observations ci-dessus poussent à considérer les variables suivantes $T_n = \dfrac{\overline{X}_n - \mu}{\sqrt{\frac{\Sigma_n^2}{n}}}$ plutôt que S_n. Le principe du test que nous allons établir repose alors sur le fait que la loi de T_n est connue :

Théorème .0

Soit $Z, X_1, \dots, X_n$ des variables aléatoires indépendantes de loi $\mathcal{N}(0,1)$. Alors :

1. $U = \displaystyle\sum_{k=1}^{n} X_k^2$ est à densité de densité : $f_U : x \longmapsto \begin{cases} c_k x^{\frac{k}{2}-1} e^{-\frac{x}{2}} & \text{si } x > 0 \\ 0 & \text{sinon} \end{cases}$

   où c_k est une constante positive (faite pour que $\displaystyle\int_{\mathbb{R}} f_U = 1$).

   On dit que U suit **la loi du χ^2 à k degrés de liberté, aussi notée χ_k^2.**
2. $\dfrac{Z}{\sqrt{\frac{U}{k}}}$ est à densité de densité $f : x \longmapsto \dfrac{d_k}{\left(1 + \frac{x^2}{k}\right)^{\frac{k+1}{2}}}$ où d_k est une constante positive (faite pour que $\displaystyle\int_{\mathbb{R}} f = 1$).

   On dit que cette variable aléatoire **suit la loi de Student à k degrés de liberté**.

On peut alors prouver [2] le résultat suivant.

Théorème .0

La variable $T_n = \dfrac{\overline{X}_n - \mu}{\sqrt{\frac{\Sigma_n^2}{n}}}$ suit la loi de Student à $n-1$ degrés de liberté.

3.3 Test de conformité à la moyenne

Sous l'hypothèse $H_0 : \mu = m$, la loi :

$$T_n = \frac{\overline{X}_n - m}{\sqrt{\frac{\Sigma_n^2}{n}}}$$

est donc connue ; une densité est donnée par :

$$f : x \longmapsto \frac{d_{n-1}}{\left(1 + \frac{x^2}{n-1}\right)^{\frac{n}{2}}}.$$

Pour un seuil α donnée, on peut donc évaluer la quantité t_α pour que :

$$\mathbb{P}(-t_\alpha \leq T_n \leq t_\alpha) = 1 - \alpha.$$

2. mais c'est très dur !

Comme :

$$\mathbb{P}(-t_\alpha \leq T_n \leq t_\alpha) = \mathbb{P}\left(m \in \left[\overline{X}_n - t_\alpha\sqrt{\frac{\Sigma_n^2}{n}}; \overline{X}_n - t_\alpha\sqrt{\frac{\Sigma_n^2}{n}}\right]\right)$$

on peut à partir de notre échantillon déterminer l'intervalle $\left[\overline{X}_n - t_\alpha\sqrt{\frac{\Sigma_n^2}{n}}; \overline{X}_n + t_\alpha\sqrt{\frac{\Sigma_n^2}{n}}\right]$ et

- si $m \notin \left[\overline{X}_n - t_\alpha\sqrt{\frac{\Sigma_n^2}{n}}; \overline{X}_n + t_\alpha\sqrt{\frac{\Sigma_n^2}{n}}\right]$, l'hypothèse $H_0 : \mu = m$ est rejetée ;
- sinon l'hypothèse est acceptée.

Travail demandé

1. Écrire une fonction `Student(L,m)` qui prend en entrée un échantillon sous forme de liste `L` et une valeur `m` et qui renvoie la réalisation de la variable T_n sous l'hypothèse $H_0 : \mu = m$.

```python
def student(L,m):
    return (moyenne(L)-m)/np.sqrt(variance(L)/len(L))
```

2. Écrire une fonction `t(alpha,n)` qui renvoie la valeur t_α pour laquelle $\mathbb{P}(-t_\alpha \leq T \leq t_\alpha) = 1 - \alpha$ pour T suivant la loi de Student à $n-1$ degrés de liberté.

```python
def t(alpha,n):
    return stats.t.ppf(1-alpha/2,df=n-1)
```

3. Recopier et compléter le programme suivant :

```python
sigma2 = rd.randint()

def EchantillonGaussien(n=5,mu=2):
        ''' renvoie une réalisation d'un n-échantillon
                d'une va de loi N(mu,sigma2)
        '''
        return rd.normal(mu,np.sqrt(sigma2),n)
```

4. Écrire une fonction `Test` avec :
   - en entrées : un échantillon sous forme de liste `L`, une valeur seuil `alpha` et la valeur `m` qu'on teste ;
   - en sortie : la valeur de T_n, l'intervalle de non-rejet de seuil `alpha` et si oui ou non l'hypothèse $H_0 : \mu =$ `m` est acceptée.

```python
def test2(L,m,alpha=0.05):
    T = student(L,m)
    talpha = t(alpha,len(L))
    inf = moyenne(L)-talpha*np.sqrt(variance(L)/len(L))
    sup = moyenne(L)+talpha*np.sqrt(variance(L)/len(L))
    if inf<=m<=sup:
        rep = "H0␣non␣rejetée"
```

```
    else:
        rep = "H0 rejetée"
    return rep, [inf,sup], T
```

5. On fixe α (par exemple $\alpha = 5\%$) et on répète `N=1000` fois l'enchaînement suivant :
   — on tire au sort un échantillon `x=EchantillonGaussien(5,2)`
   — on test l'hypothèse $H_0 : \mu = 2$.

```
def verification(alpha=0.05, N=1000,n=5,mu=2):
    s = 0
    for k in range(N):
        x= test(EchantillonGaussien(n ,mu),mu,alpha)
        if x[0] == "H0 rejetée":
            s+=1
    return s/N
```

La fonction `verification` donne le pourcentage de fois où l'on rejette H_0 alors qu'elle est vérifiée. Comme on utilise une zone de rejet avec un risque de 5%, `verification` doit renvoyé une valeur d'environ 0.05 ce qui est confirmé par les simulations.

Annexe

Lois usuelles à densité

Définition .1

- Soit μ et $\sigma > 0$ deux réels. On dit qu'une variable aléatoire X suit la **loi normale de paramètres μ et σ^2** et on note $X \hookrightarrow \mathcal{N}(\mu, \sigma^2)$ si X a pour densité la fonction f définie par

$$\forall x \in \mathbb{R}, \quad f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

- Soit $\lambda > 0$. On dit qu'une variable aléatoire X suit la loi **exponentielle de paramètre** $\lambda > 0$ et on note $X \hookrightarrow \mathcal{E}(\lambda)$ si X a pour densité la fonction f définie par

$$\forall x \in \mathbb{R}, \quad f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases}.$$

3.4 Simulations de lois usuelles

En important le module `numpy.random` avec la commande :

`import numpy.random as rd`

on peut simuler les lois usuelles :

— `rd.exponential(lambda,n)` : simule un n échantillon de loi $\mathcal{E}(\lambda^{-1})$,

— `rd.normal(mu,sigma,n)` : simule un n échantillon de loi $\mathcal{N}(\texttt{mu},\texttt{sigma}^2)$,
— `rd.poisson(lambda,n)` : simule un n échantillon de loi $\mathcal{P}(\lambda)$,
— ...

En important le module `matplotlib.pyplot` avec la commande :

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

la commande

```
hist(L, density = True)
```

permet de tracer l'histogramme des fréquences de la liste `L`.

Loi de Student

En important le module `scipy.stats` avec la commande :

```
import scipy.stats as stats
```

la commande `stats.t.ppf(a,df =n)` donne la valeur t_a pour lequel on a

$$\mathbb{P}(T \leq t_a) = a$$

pour T suivant une loi de Student à n degrés de liberté.
Comme la densité donnée ci-dessus est paire, on a :

$$\mathbb{P}(T \leq -t) = \mathbb{P}(T \geq t)$$

donc

$$\mathbb{P}(-t \leq T \leq t) = \mathbb{P}(T \leq t) - \mathbb{P}(T \leq -t) = 2\mathbb{P}(T \leq t) - 1.$$

Pour $a = 1 - \frac{\alpha}{2}$, on a alors :

$$\mathbb{P}(-t_a \leq T \leq t_a) = 2a - 1 = 1 - \alpha.$$