Feuille de TP Python 09

Théorèmes limites en probabilité, Statistiques inférentielles

1 Introduction

L'objet de ce TP est d'une part de présenter mathématiquement les théorèmes limite de probabilités au programme et d'autre part, d'expérimenter et visualiser informatiquement ces résultats.

La dernière partie est consacrée à illustrer l'utilisation en statistiques.

2 Les théorèmes limite au programme de BCPST

2.1 Notion de convergence en loi

Définition 1. *Soit $(X_n)_{n\in\mathbb{N}}$ une suite de variables aléatoires réelles, de fonctions de répartitions respectives F_n. Soit X une v.a.r de fonction de répartition F. On dit que $(X_n)_{n\in\mathbb{N}}$ converge en loi vers X si :*

$$\forall x \in \mathbb{R}, F\ \mathscr{C}^0 \text{ en } x \Rightarrow \left(F_n(x) \overset{n\to+\infty}{\to} F(x)\right)$$

Proposition 2. *Si $(X_n)_{n\in\mathbb{N}}$ est une suite de v.a. prenant des valeurs entières positives. $(X_n)_{n\in\mathbb{N}}$ converge en loi vers X, à valeurs entières, si et seulement si*

$$\forall k \in \mathbb{N}, \left(\mathbb{P}(X_n = k) \overset{n\to+\infty}{\to} \mathbb{P}(X = k)\right)$$

Nous avons déjà rencontré une telle convergence en loi dans le théorème liant loi binomiale et loi de POISSON que nous rappelons ici, avec ce nouveau vocabulaire :

Théorème 3: Approximation Binomiale–POISSON

Soit $\lambda \in]0,+\infty[$ et $(p_n)_{n\in\mathbb{N}^*} \in]0,1[^{\mathbb{N}^*}$ telle que $p_n \sim_{n\to+\infty} \frac{\lambda}{n}$. Si $X_n \sim \mathscr{B}(n, p_n)$ alors $(X_n)_{n\in\mathbb{N}}$ converge en loi vers X, une variable de POISSON, de paramètre λ.

Démonstration. Il s'agit de démontrer que, $\forall k \in \mathbb{N}$,

$$\binom{n}{k} p_n^k.(1-p_n)^{n-k} = \frac{\lambda^k}{k!}.\underbrace{\frac{n.(n-1)\dots.(n-k+1)}{n^k}(1+o(1))^k.(1-p_n)^k}_{\to 1 \text{ pk ?}}.\underbrace{\left(1-\frac{\lambda}{n}+o\left(\frac{1}{n}\right)\right)^n}_{\to e^{-\lambda} \text{ pk ?}} \overset{n\to+\infty}{\to} \frac{\lambda^k}{k!}.e^{-\lambda}$$

□

Exercice 1.— Dans un script nommé `Poisson.py`, après importation de `numpy`, `numpy.random`, `numpy.matplotlib.pyplot`
1. Ecrire une fonction d'entête `Binomiale(n,p)` retournant une valeur simulée d'une v.a. binomiale $\mathscr{B}(n,p)$
2. Ecrire une fonction d'entête `Poisson(lambada)` retournant une valeur simulée d'une v.a. de POISSON $\mathscr{P}(\lambda = \texttt{lambada})$.
Indication:Simuler une binomiale $\mathscr{B}(n,\lambda/n)$ avec $n = 100(\lfloor\lambda\rfloor+1)$. Quelle espérance, quelle variance ?
3. Ecrire une fonction d'entête `HistoPoisson(lambada, NS = 10000)` qui tire au sort `NS` valeurs suivant une loi de POISSON de paramètre $\lambda = \texttt{lambada}$ et en trace l'histogramme.

Exécuter et observer !

NB : Pour les exercices de ce TP ;

- **L'histogramme cumulé et normalisé d'une liste de nombres `liste` s'obtient (sur le canevas courant) à l'aide de `plt.hist(liste, density = True, cumulative = True)`.**
- **La fonction de répartition d'une loi $\mathscr{N}(0,1)$, est encodée en Python/Numpy par `norm.cdf` après importation de `norm` depuis `scipy.stats`. On peut aussi y placer des paramètres m et σ^2, voir `help(norm.cdf)`, paramèter `loc` et `scale`.**

2.2 Loi du min, loi du max

Exercice 2.—Soit $(T_n)_{n\in\mathbb{N}^*}$, une suite de v.a. indépendantes, de loi exponentielle de paramètre λ. Pour $N \in \mathbb{N}^*$ donné, on pose $M_N = \max(T_1, \ldots, T_N)$. Les fonctions demandées sont à placer dans un script nommé `Gumbel.py`.
1. Montrer que M_N est une v.a à pour fonction de répartition la fonction F_{M_N} donnée par :

$$\forall t \in \mathbb{R},\, F_{M_N}(t) = \begin{cases} (1-e^{-\lambda.t})^N & \text{si } t > 0 \\ 0 & \text{sinon} \end{cases}$$

2. Donner la fonction F_N de répartition de $M_N - \frac{\ln N}{\lambda}$
3. Ecrire [1] une fonction `MaxExp(N,lambada)` retournant une valeur aléatoire tirée suivant la loi de M_N.
4. Calculer $F(x)$, la limite de $F_N(x)$ lorsque $N \to +\infty$, pour chaque $x \in \mathbb{R}$.
5. La fonction F ainsi définie est-elle la f.r. d'une v.a. X à densité ? Quelle est la fonction des quantiles de X ?
6. Ecrire une fonction `Gumbel(lambada)` retournant une valeur aléatoire tirée suivant la loi de X.
7. En exhibant des histogrammes « cumulés » calculés à partir d'un (grand) nombre `NS` de simulations de `MaxExp` et `Gumbel`, illustrer la convergence en loi de $M_N - \frac{\ln(N)}{\lambda}$ vers X.

2.3 TCL : Le théorème central limite/de la limite centrale

Si $(X_n)_{n\in\mathbb{N}^*}$ est un échantillon d'une v.a. réelle X, on note, pour tout $N \in \mathbb{N}^*$, la N-ième moyenne empirique (deux notations) et la N-ième variance empirique, les v.a. :

$$\overline{X}_N = M_N = \frac{1}{N}\sum_{n=1}^{N} X_n \text{ et } S_n^2 = \frac{1}{N}\sum_{n=1}^{N}(X_n - M_N)^2 \overset{\text{KOENIG–HUYGHENS}}{=} \frac{1}{N}\sum_{n=1}^{N} X_n^2 - M_N^2.$$

Proposition 4. — *Si X admet une espérance μ alors, pour tout $N \in \mathbb{N}^*$, $\overline{X}_N$ admet pour espérance μ.*
— *Si X admet une variance σ^2 alors, pour tout $N \in \mathbb{N}^*$, $\overline{X}_N$ admet pour variance $\frac{\sigma^2}{N}$.*
— *Si X admet une variance σ^2 alors, pour tout $N \in \mathbb{N}^*$, la cenbtrée-réduite de $\overline{X}_N$ est*

$$\overline{X}_N^* = \sqrt{N}\frac{\overline{X}_N - \mu}{\sigma} = \frac{1}{\sqrt{N}\sigma}.\left(\sum_{n=1}^{N}(X_n - \mu)\right) = \frac{\left(\sum_{n=1}^{N} X_n\right) - \mu}{\sqrt{N}\sigma}.$$

Théorème 5: forme classique du TCL : P. Lévy, 1935/Admis

Si X admet une variance, $(X_n)_{n\in\mathbb{N}^*}$ est un échantillon de X – une suite *indépendante*, de v.a. distribuées comme X– alors la suite $(\overline{X}_N^*)_N$ des moyennes empiriques centrées-réduites converge en loi vers une variable $Z \sim \mathcal{N}(0,1)$.

Exercice 3.—Reprendre la fonction `Binomiale` de l'exercice 1 et la placer dans un script nommé `MoivreLaplace.py`.

Ecrire une fonction d'entête `ComparaisonGraphique(n,p,NS = 10000)` qui tire un (grand) nombre NS de valeurs simulées de `Binomiale(n,p` et trace, sur un même graphique l'histogramme cumulé de ces valeurs et le graphe de la fonction de répartition d'une loi normale $\mathcal{N}(n.p, n.p(1-p))$.

Exécuter et observer (varier les valeurs de n et p :plus p est proche de 0 ou 1, plus n doit être grand) !

Exercice 4.—Douze uniformes. Dans un fichier nommé `SU12.py`.
1. Définir une fonction Python d'entête `SU12()` retournant une valeur simulée de la somme–*centrée* de 12 v.a. indépendantes uniformes sur $[0,1]$.
2. Ecrire une fonction d'entete `ComparaisonGraphique(NS = 10000)` qui tire un (grand) nombre NS de valeurs simulées de `SU12()` et trace, sur un même graphique l'histogramme cumulé de ces valeurs et le graphe de la fonction de répartition d'une loi normale $\mathcal{N}(0,1)$.

Exercice 5.— Dans un fichier nommé `TCL.py`.
1. Ecrire une fonction d'entête `Xpref()` simulant une v.a. réelle (finie, discrète, à densité, autre ?) de votre loi préférée centrée, réduite. L'entête peut comporter des arguments optionnels avec valeurs par défaut.
2. Ecrire une fonction d'entête `Xbarre_etoile(N = 15, X = Xpref)` retournant une valeur simulée de la N-ième moyenne empirique centrée réduite d'un échantillon de v.a. X simulée par la fonction Python `X` passée en argument.
3. Ecrire une fonction d'entete `ComparaisonGraphique_etoile(N, NS = 100_000)` qui tire un (grand) nombre NS de valeurs simulées de `Xbarre_etoile` et trace, sur un même graphique l'histogramme cumulé de ces valeurs et le graphe de la fonction de répartition d'une loi normale $\mathcal{N}(0,1)$.

On conseille par ailleurs d'ajouter comme option `bins = int(NS/100)` à la commande `plt.hist` pour avoir suffisamment de finesse.

Exécuter (faire augmenter les valeurs de `N`) et observer !

1. NB : Vous pouvez, si vous voulez simplifier, supposer que $\lambda = 1$ et enlever l'argument `lambada` des fonctions. Ce n'est pas là ue réside la difficulté.

Théorème 6: 2e forme–statistique– du TCL

Si X admet une variance, $(X_n)_{n\in\mathbb{N}^*}$ est un échantillon de X – une suite *indépendante*, de v.a. distribuées comme X– alors la suite $(\overline{X}_N^{**})_N$converge en loi vers une variable $Z \sim \mathcal{N}(0,1)$ où

$$\forall N \in \mathbb{N}^*, \overline{X}_N^{**} = \sqrt{N}\frac{\overline{X}_N - \mu}{S_N}.$$

Exercice 6.—Reprendre l'exercice 5 et compléter le script avec un travail du même type que celui fait sur $\overline{X}_N^*$ simulée par `Xbarre_etoile` mais concernant $\overline{X}_N^{**}$ simulée par `Xbarre_etoile2`

Exercice 7.—

1. Dans un script nommé `EchantillonConformiteMoyenne.py` : Simuler[2] $N = 50$ (ou autre) valeurs votre v.a. préférée par une fonction `X()` et placer ces 50 valeurs dans un vecteur `Numpy` de *shape* `(N,)` `VX`.

Sauver ce tableau `VX` par la commande `np.save("VX.npy",VX)`

2. Dans un script `ConformiteMoyenne.py`, **2.a.** charger le tableau `VX` par la commande `VX = np.load("VX.npy")` puis :

2.b. Ecrire une fonction `XNbarre_etoile2(mu)` qui évalue (et retourne) la valeur observée de $\overline{X}_N^{**}$ en supposant que les nombres présents dans `VX` sont des valeurs observée d'un échantillon d'une v.a. X admettant pour espérance $\mu = (mu)$.

2.c. Ecrire une fonction `pvaleur(mu)` qui retourne–après l'avoir évaluée en utilisant la fonction de répartition d'une v.a. normale centrée réduite, la valeur de la probabilité d'observer la valeur retournée par `XNbarre_etoile2(mu)` ou une valeur encore moins susceptible d'être observée.

Plus clairement : connaissant $z =$ `XNbarre_etoile2(mu)`, retourner la valeur de $\mathbb{P}(Z \in]-\infty, -|z|[\cup]+|z|, +\infty[)$ où $Z \sim \mathcal{N}(0,1)$.

2.d. Tester (en cherchant à le maximiser) `pvaleur(mu)` avec différentes valeurs de `mu`. Quelle valeur de `mu` trouvez vous ? Ccl ?

2.e. Pour quelle plage de valeurs de `mu`, `pvaleur(mu)` est il supérieur à 0.05 ? Vérifier qu'il s'agit de l'intervalle $\overline{X}_N \pm 1.96.S_N/\sqrt{N}$. (On pourra faire imprimer les bornes de cet intervalle par la fonction `XNbarre_etoile2(mu)`). La vraie valeur de l'espérance de X est-elle dans cet intervalle[3] ?

3. Faire la même opération avec le fichier `VX.npy` de votre voisin/voisine. Quelle valeur de `mu` trouvez vous ? Ccl ?

3 Lois faibles des grands nombres

Théorème 7: Loi faible des grands nombres, version générale/admise

Si X admet une espérance, $(X_n)_{n\in\mathbb{N}^*}$ est un échantillon de X – une suite *indépendante*, de v.a. distribuées comme X– alors, de façon équivalente :

— la suite $(\overline{X}_N)_N$ des moyennes empiriques converge en loi vers une variable C, constante égale à μ ;

— $\forall \varepsilon > 0$,

$$\mathbb{P}\left(\left|\overline{X}_N - \mu\right| > \varepsilon\right) \overset{N\to+\infty}{\to} 0.$$

Remarque : l'équivalence n'est pas si évidente et est basée sur les deux faits suivants :

— la fonction de répartition d'une v.a. constante valant μ est la fonction F :

$$\forall x \in \mathbb{R}, F(x) = \begin{cases} 0 & \text{si } x < \mu \\ 1 & \text{si } x \geq \mu \end{cases}$$

— Si F_N désigne la fonction de répartition de $\overline{X}_N$, pour $\varepsilon > 0$,

$$F_N(\mu - \varepsilon) = \mathbb{P}(\overline{X}_N - \mu \leq -\varepsilon) \text{ et } 1 - F_N(\mu + \varepsilon) = \mathbb{P}(\overline{X}_N - \mu > +\varepsilon)$$

Exercice 8.—Expliquer pourquoi (donner dans chaque cas les v.a. dont on évaluer les espérances par ce procédé) ce théorème est à la base des méthodes de simulation permetant d'évaluer certaines caractéristique de la loi d'une v.a. X par calcul de moyenne sur un grand nombre de simulations dans les cas suivants :

1. Calcul approché de l'espérance de X.

2. Calcul approché de la variance de X.

3. Calcul approché de la probabilité d'un événement du type $\{X \in A\}$.

4. Calcul approché de la fonction de répartition de X par le calcul d'une fonction de répartition d'un échantillon (histogramme cumulé).

5. Calcul approché de la distribution de X par le calcul d'un histogramme.

2. faire en sorte (transformation affine des valeurs) que la v.a. simulée ait une espérance « peu commune », du genre π, $\ln 2$ ou $53,53$.

3. s'il n'y est pas, c'est possible MAIS vous faites partie des 5% de gens qui n'ont pas de chance dans la vie, ne tentez pas de jouer au Loto, vous n'avez aucune chance de gagner le remboursement du ticket

Théorème 8: Loi faible des grands nombres, version variance/vitesse

Si X admet une *variance*, $(X_n)_{n\in\mathbb{N}^*}$ est un échantillon de X alors :

$$\forall\varepsilon>0, 0\leq\mathbb{P}\left(\left|\overline{X}_N-\mu\right|>\varepsilon\right)\leq\frac{\mathbb{V}(X)}{\varepsilon^2.N}\overset{N\to+\infty}{\to}0.$$

Exercice 9.—

1. Montrer que $f : x\mapsto\frac{2}{x^3}1\!\!1_{\{x\geq1\}}$ est une densité de probabilité et donner la fonction de répartition F d'une v.a. X ayant cette densité [4]. Calculer l'espérance de X et donner une fonction des quantiles de X.

2. Dans un script nommé `LFGN-essai-Pareto.py`.

2.a. Ecrire une fonction `X` retournant une valeur simulée de X.

2.b. Ecrire une fonction `EvalEX(N)` retournant la moyenne d'un tirage d'un N-échantillon de X.

2.c. Ecrire une fonction `ProbaEcart(N,epsilon,NS = 10000)` retournant la proportion, sur `NS` essais, de tirages de `EvalEX(N)` s'écartant – en valeur absolue– de plus de `epsilon` de la valeur théorique de l'espérance de X.

3. Fixer $\varepsilon=1$, tracer le graphe de $N\mapsto N.$`ProbaEcart(N,1)` pour N variant de 10 à 100. Commentaire ?

4. Recommencer avec la fonction $f : x\mapsto\frac{k}{x^{k+1}}1\!\!1_{\{x\geq1\}}$ avec différentes valeurs de k entier. Commentaire ?

5. Et avec X une variable $\mathscr{E}(1)$?

4. Une telle v.a. est appelée une v.a. de PARETO, c'est utile, parait-il, en économie