

Informatique – TP 6 – Modélisation de lois statistiques

Exercice 1 – On veut écrire des fonctions dont la sortie est une variable aléatoire suivant certaines lois classiques ; les paramètres des lois seront également des paramètres de la fonction.

Exemple : Pour modéliser la loi de Bernoulli de paramètre p , on crée une fonction prenant en paramètre la valeur p , et renvoyant soit 0, avec une probabilité de $1 - p$, soit 1, avec une probabilité de p . La seule loi que connaisse *a priori* Turbo Pascal est la loi uniforme (c'est la fonction `random`). Pour modéliser la loi de Bernoulli, on procède alors de la sorte.

Supposons que nous voulions approcher la valeur de p à 10^{-2} près. Alors, on commence par tirer aléatoirement (de manière uniforme) un entier n compris entre 1 et 100.

La probabilité de tirer un nombre $n \leq 100p$ (donc un entier de $\llbracket 1, E(100p) \rrbracket$) est $\frac{E(100p)}{100}$, c'est-à-dire p tronqué à 10^{-2} près. La probabilité de tirer un nombre $n > 100p$ est $1 - \frac{E(100p)}{100}$. Ainsi, à une erreur près de 10^{-2} sur p , on obtient une loi de Bernoulli de paramètre p en donnant à la fonction la valeur 1 si $n \leq 100p$ et 0 si $n > 100p$.

Écrire des fonctions suivant les lois suivantes, et calculer sur des exemples la moyenne obtenue en répétant 1000 fois l'expérience :

1. La loi uniforme $\mathcal{U}(\llbracket n, m \rrbracket)$.
2. La loi de Bernoulli $\mathcal{B}(1, p)$ (on prendra une approximation de p à 10^{-4})
3. La loi binomiale $\mathcal{B}(n, p)$ (même remarque)
4. La loi géométrique $\mathcal{G}(p)$ (même remarque)
5. La loi de Pascal $\mathcal{P}(r, p)$ (idem)
6. La loi de Poisson de paramètre $\lambda \leq 10$ (l'approcher par une loi convenable ; prendre $n = 100$)
7. La loi hypergéométrique $\mathcal{H}(N, n, p)$ (on entrera comme paramètre non pas p , mais le nombre entier Np , nombre de boules blanches).
Indication : Avoir deux variables indiquant le nombre de boules blanches et noires restantes. Tirer un nombre inférieur au nombre total de boules restantes ; décider si c'est une boule blanche ou une boule noire suivant la valeur.
8. Temps d'attente dans un tirage sans remise (paramètres N, Np).

Remarque importante : Nous sommes obligés dans cet exercice de prendre une valeur approchée de p , car nous voulons laisser le choix à l'utilisateur de la valeur de p . Si la valeur de p est fixée à l'avance (et apparaît donc en dur dans le programme), il n'est souvent pas nécessaire de faire cette approximation. En effet, dans la plupart des cas, p est donné sous forme d'une fraction $p = \frac{a}{b}$, où a et b sont des entiers positifs, $a < b$. Alors la probabilité de tirer un nombre $n \leq a$ lorsqu'on tire aléatoirement n dans $\llbracket 1, b \rrbracket$ de manière uniforme est *exactement* p .

Exercice 2 –

1. Écrire un programme demandant à l'utilisateur un entier strictement positif r (test de compatibilité à faire) et renvoyant les valeurs d'une variable aléatoire suivant la loi du temps d'attente de la première série de r « six » successifs dans une succession de lancers de dés (non pipés). Pour chaque valeur de r choisie, donner une valeur approchée de l'espérance (prendre la moyenne sur 1000 expériences).
2. Écrire un programme renvoyant les valeurs d'une variable aléatoire suivant la loi du temps d'attente de la première occurrence de PPFPPF dans une suite de lancers mutuellement indépendants d'une pièce équilibrée.
3. Même question avec PPFPPFPFFPPF.