TP d'initiation à MATLAB n^0 5 : Simulation de variables aléatoires. Statistique descriptive

On entrevoit des méthodes pour simuler des réalisations de variables aléatoires suivant les principales lois de probabilités. On pourra alors générer des données auxquelles seront appliquées des statistiques descriptives.

Simulation de variables aléatoires

A partir du générateur de nombres pseudo-aléatoires décrit dans le TP précédent (fournissant approximativement des réalisations de variables uniformes indépendantes sur $[0, 1]$), on peut simuler des réalisations de variables aléatoires discrètes (Bernoulli, binomiale,...), mais aussi absolument continues (normales, exponentielles, Cauchy,...).

Simulation d'une variable aléatoire discrète : Bernoulli

On veut générer une réalisation d'une variable de Bernoulli X telle que $P(X = 1) = p$ et $P(X = 0) = 1 - p$, avec $p \in [0, 1]$. Que se passe-t-il si $p = 1$? Quelles sont les valeurs possibles pour X ?
Quelle est la fonction de répartition de X $F_X(x) = P(X \leq x)$?
Créer alors une fonction (donc un M-fichier) appelée *Bernoulli*.

`% Simulation Bernoulli`	Commentaire descriptif de la fonction.
`function x = Bernoulli(p)`	Le paramètre d'entrée est $p \in [0, 1]$.
`x = 0;`	On initialise x, réalisation de X.
`u = rand;`	On génere une réalisation u d'une variable uniforme U sur $[0, 1]$
`if (u >= p)`	Test sur u.
`x = 1;`	Dans le cas où $u < p$, on reste avec $x = 0$.
`end`	Fin du test.
	Sauvegardez la fonction, puis revenez dans la fenêtre de commande.
`Bernoulli(0.3)`	Un essai. Faire d'autres essais. Comment faire pour vérifier empiriquement que l'on a bien généré une variable de Bernoulli ? Montrer théoriquement que c'est bien le cas (pour cela, il suffit de calculer F_X en posant $X = 1$ si $U \geq p$, et $X = 0$ sinon). Modifier votre fonction de façon à générer maintenant n réalisations indépendantes de X. Pour cela, mettre n comme paramètre d'entrée et utiliser $rand(1, n)$...

Simulation d'une variable aléatoire discrète quelconque

Ouvrir un nouveau M-fichier appelé *VAdiscrete.m*.

`% Generateur d'une v.a.`	On génere une réalisation d'une v.a. Y qui suit une loi discrète
`% a valeurs dans {0,1,2}`	à valeurs dans $\{0, 1, 2\}$ étant données les probabilités d'avoir 0, 1 et 2.
`function [y] = VAdiscrete(p0,p1)`	On construit une fonction appelée *VAdiscrete* dépendant de 2 paramètres: $p0$ est la probabilité d'avoir 0, et $p1$ celle d'avoir 1.
`u = rand; y = 0;`	Comme précédemment...
`if (u >= p0) & (u < p0+p1)`	Suivant la valeur de u, y pourra être égale à 0 (par défaut),
`y = 1;`	ou 1,
`elseif (u >= p0+p1)`	
`y = 2;`	ou 2.
`end`	Il faut fermer chaque conditionnement.
`end`	Sauver le fichier et quitter. Faire des essais.
	Transformer la fonction pour simuler n réalisations indépendantes de Y.

Exercices

1. Suivant la procédure précédente, simuler k réalisations indépendantes d'une v.a. suivant une loi binomiale $\mathcal{B}(5, 0.3)$. N'y-aurait-il pas un autre moyen de simuler une telle variable (penser à une somme de v.a.) ? En déduire une fonction simulant k réalisations indépendantes d'une v.a. suivant une loi binomiale $\mathcal{B}(n, p)$, où n et p sont des paramètres.

2. Simuler une réalisation d'une variable aléatoire X de Poisson de paramètre $\lambda > 0$ (on rappelle qu'alors $P(X = k) = \dfrac{\lambda^k}{k!}e^{-\lambda}$ pour $k \in \mathbb{N}$). A quels problèmes êtes-vous confrontés ?

Simulation de variables aléatoires (suite et fin)

Un peu de théorie d'abord : on suppose que U est une variable aléatoire de loi uniforme sur $[0, 1]$. Soit X une variable aléatoire absolument continue de fonction de répartition F_X telle que F_X est une fonction continue strictement croissante de I (intervalle de $\mathbb{R}$) dans $]0, 1[$. Alors, sur $]0, 1[$, F_X admet une application réciproque F_X^{-1} $:]0, 1[\mapsto I$. Or U étant une variable uniforme sur $[0, 1]$, on a $F_X(x) = P(U \leq F_X(x)) = P(F_X^{-1}(U) \leq x)$ pour $x \in I$. En conséquence, X et $F_X^{-1}(U)$ ont la même fonction de répartition donc la même distribution (travailler avec $]0, 1[$ ou $[0, 1]$ ne change rien) et ainsi, pour simuler une réalisation de X, il nous suffira de calculer $F_X^{-1}(u)$ où u est une réalisation de U (donc un nombre pseudo-aléatoire). Cette méthode peut-être généralisée grâce à la notion de réciproque généralisée, et l'on retrouve alors également la simulation de variables discrétes.

———————————————————————————

Simulation de variables aléatoires exponentielles

———————————————————————————

On rappelle qu'une variable aléatoire X de loi exponentielle et de paramétre β est telle que $F_X(x) = 0$ si $x < 0$ et $F_x(x) = 1 - e^{-\beta x}$ si $x \geq 0$. Créer maintenant une fonction permettant de générer n réalisations d'une variable aléatoire X de loi exponentielle et de paramétre β.

———————————————————————————

Simulation de variables aléatoires gaussiennes

———————————————————————————

La méthode proposée précedemment nécessite la connaissance explicite de F_X^{-1}. Comment faire si cette fonction ne peut être exactement exprimée ? Ce problème se pose pour une variable gaussienne centrée réduite $\mathcal{N}(0, 1)$ réduite (de loi $\mathcal{N}(0, 1)$) qui est telle que $F_X(x) = \displaystyle\int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$ pour $x \in \mathbb{R}$: il n'y a pas de formulation explicite de la réciproque d'une telle fonction.

$randn(1, 10)$	Simulation de n réalisations $\mathcal{N}(0, 1)$. Il existe donc dèjà un générateur. Nous allons retrouvé cette fonction. Ouvrir *Gauss.m*.
% Generateur de v.a.	Descriptif...
% de loi $N(0, 1)$	
function $[x, y] = Gauss$	On construit une fonction appelée *Gauss* qui simule 2 réalisations x et y de 2 v.a. indépendantes X et Y de loi $\mathcal{N}(0, 1)$.
$(u1, u2) = rand(2, 1);$	On simule 2 nombres pseudo-aléatoires.
$v1 = sqrt(-2 * log(u1))$	On simule une réalisation d'une v.a. de loi exponentielle de paramétre 0.5.
$x = v1 * sin(2 * pi * u2)$	Simulation de x;
$y = v1 * cos(2 * pi * u2)$	Simulation de y;
	Sauver le fichier et quitter. Faire des essais.

L'idée de cette simulation est le fait que dans le plan cartésien, si on considère un point M de coordonnées (X, Y), l'angle $\widehat{Ox, OM}$ suit une loi uniforme sur $[0, 2\pi]$ et la distance OM suit une loi exponentielle de paramétre 0.5.

Exercices

1. A partir de la fonction de répartition de cette variable aléatoire, simuler des réalisations d'une v.a. de Cauchy dont la densité de probabilité est : $f(x) = \dfrac{1}{\pi(1+x^2)}$ (créer une fonction $VACauchy.m$).

2. Simuler des réalisations d'une loi unfiorme sur $[a, b]$ avec ($a < b$ quelconques, et d"une loi normale $\mathcal{N}(m, \sigma^2)$ avec m et σ^2 quelconques.

3. **(Facultatif)** Simuler une réalisation d'une variable aléatoire $\mathcal{N}(0, 1)$ en utilisant une fonction de répartition réciproque approchée calculée à partir de l'approximation d'une intégrale par Matlab.

Statistique descriptive

Nous nous sommes intérrogés un peu plus haut sur la possibilité de vérifier empiriquement la validité d'une simulation d'une variable aléatoire. Avant d'en arriver là, voyons les quelques notions élémentaires de statistique descriptive proposées par Matlab

$clear$	On efface tout.
$Y = rand(34, 1)$	On génère un vecteur colonne composé de 34 réalisations indépendantes.
$hist(Y)$	Effectue le tracé d'un histogramme des y_i. Sans option, l'histogramme est généré automatiquement avec 10 classes de même étendue.
$hist(Y, 14)$	On obtient un histogramme avec 14 classes et non 10 obtenu par défaut.
$mean(Y)$	Calcule la moyenne des y_i.
$std(Y)$	Calcule l'écart-type des y_i.
$var(Y)$	Calcule la variance des y_i.
$median(Y)$	Détermine la médiane des y_i.
$prctile(Y, 0.2)$	Détermine le quantile à 20% des y_i. Faire d'autres essais.
$boxplot(Y)$	Graphe appelé encore “boîte à moustaches”. Regarder le help pour comprendre.
$X = (1967 : 2000)'$	On génére un vecteur colonne de longueur 34. Ce vecteur représente par exemple des années.
$Z = 0.1 * X + 2 + Y$	On construit ainsi une nouvelle série de données.
$plot(X, Z, '*')$	Représentation du nuage de points, montrant l'évolution de Z en fonction des années.
$lsline$	Représente directement la droite de régression linéaire de Z par X.
$S = cov(X, Z)$	Représente la matrice de covariance empirique des x_i et des z_i. Plus précisément,
$S(1, 1)$	Représente la variance des x_i.
$S(2, 2)$	Représente la variance des z_i.
$S(1, 2)$	Représente la covariance de x_i et des z_i (même chose pour $S(2, 1)$).
$R = corrcoef(X, Z)$	Représente la matrice de corrélation empirique des x_i et des z_i.
$R(1, 2)$	Représente le coefficient de corrélation linéaire des x_i et z_i.
$c = polyfit(X, Z, 1)$	Calcule directement les coefficients de la droite de régression de Z par X. Cette droite a ainsi pour équation $z = c(1)x + c(2)$. Vérifier sur le graphe en la traçant à nouveau à partir de cette équation.

Exercices

1. Faire un programme permettant de tracer le polygône des fréquences cumulées (ou la fonction de répartition empirique) de Y. On pourra par exemple utiliser la fonction *sort*... Retrouver sur ce graphe, la médiane et les quantiles précédents.

2. Faire la régression linéaire de Y par X et expliquer le résultat. De même, faire la régression linéaire de Z par Y et expliquer.