

Introduction aux probabilités - Licence MIA 2e année - parcours Informatique
Contrôle continu du 14/01/2013 - Durée : 1 heure

Les documents ne sont pas autorisés. **Tous les codes demandés devront être écrits très soigneusement afin que chaque caractère soit lisible.**

Exercice 1

1. Ecrire une fonction **Esperance** qui calcule l'espérance d'une variable aléatoire à support fini. La fonction prendra en entrée deux vecteurs **x** et **p**, où **x** représente un vecteur contenant les valeurs x_i prises par la variable, et **p** un vecteur contenant les probabilités $p_i = P(X = x_i)$ correspondantes.
2. Ecrire une fonction **Variance** à deux arguments **x** et **p** qui calcule la variance d'une variable aléatoire à support fini, en faisant appel à la fonction **Esperance**.

Exercice 2 Pour chacune des commandes R suivantes, dire à quelle loi de probabilité elle correspond et ce qu'elle renvoie, puis donner la commande prédéfinie de R équivalente :

1. `sum(sample(c(0,0,1),15,rep=TRUE))`
2. `exp(-3)*3^4/factorial(4)`
3. `sum(runif(10)<0.7)`
4. `3*exp(-3*4)`

Exercice 3 Un système numérique transmet des données sous forme d'un message de n bits (valeur 0 ou 1). Afin de repérer les erreurs de transmission, un $(n + 1)^e$ bit appelé *bit de parité* est ajouté à la fin du message avant la transmission : si le nombre de 1 dans les n premiers bits est pair, le bit de parité est mis à zéro, sinon il est mis à 1. Lors de la transmission, chacun des $n + 1$ bits peut être altéré (sa valeur passe de 0 à 1 ou inversement), avec probabilité p et de façon indépendante des autres bits. Enfin en réception, la système signale une erreur lorsque le nombre de bits à 1 parmi les n premiers bits est en accord avec la valeur du bit de parité.

1. Ecrire une fonction **AjoutBitParite** prenant en entrée un vecteur **message** supposé ne contenir que des 0 et des 1 et renvoyant un vecteur **paquet** de taille $n + 1$ (où n est la taille de **message**) composé du même vecteur et du bit de parité en dernière position. On pourra utiliser l'opération modulo, notée `%%` en R (exemple : `8 %% 3` renvoie 2, résultat de 8 modulo 3).
2. Ecrire une fonction **SimulTransmission** prenant en entrée un vecteur **paquet** et un scalaire **p**, et renvoyant le paquet où chacun des $n + 1$ bits a été altéré avec probabilité p .
3. Ecrire une fonction **Controle** prenant en entrée un vecteur **paquet** qui vérifie le paquet en réception en comparant la parité du nombre de valeurs à 1 dans les n premiers bits avec la valeur du bit de parité, et renvoie **TRUE** si le paquet semble correct, **FALSE** sinon.

Annexe : Loix prédéfinies en R

Loi	Appellation	Arguments	Valeurs par défaut
Binomiale de paramètre $(n, q) \in \mathbb{N}^* \times [0, 1]$	binom	size=n prob=q	
Géométrique de paramètre $q \in [0, 1]$	geom	prob=q	
Normale (ou Gaussienne) de paramètre $(\mu, \sigma) \in \mathbb{R} \times \mathbb{R}_+$	norm	mean= μ sd= σ	mean=0 sd=1
Poisson de paramètre $\lambda \in \mathbb{R}_+$	pois	lambda= λ	
Uniforme sur $[min, max]$	unif	min= <i>min</i> max= <i>max</i>	min=0 max=1
Exponentielle de paramètre $a > 0$	exp	rate= <i>a</i>	rate=1