

Statistiques TP 1

Probabilités et statistiques avec Maple

Pour lancer maple10, tapez xmaple10c, ou xmaple10 -cw. Cela permet de l'avoir en mode classique, plus similaire à maple : à l'agrégation, seul maple7 est pour l'instant disponible. On utilisera donc avec maple10 des packages existant aussi sous maple7. **N'oubliez pas de sauver régulièrement !!!**

1 Premiers pas sous maple

Les commandes de maple se terminent toujours par un point virgule (;), ou par deux points (:) si on ne veut pas que maple affiche de résultats. Pour affecter quelque chose à un nom, on utilise := (et pas =). Ainsi $a := 2$; affecte la valeur 2 à a . N'ayez pas peur de donner un nom à plupart des calculs que vous ferez, cela permet d'y faire référence.

A la différence du logiciel **matlab** où la structure de base est le tableau, pour **maple** la structure omniprésente est plutôt la suite (ou séquence, *sequence* en anglais).

1.1 Séquences

Une séquence est une énumération d'objets séparés par des virgules. Ainsi **1, toto, x+y, 2.3** est une séquence. Si s_1 et s_2 sont deux séquences, $s_3 := s_1, s_2$ fournit la séquence obtenue en les mettant à la suite.

Pour obtenir le i -ème élément d'une séquence s , on utilise **s[i]**.

Une commande extrêmement importante est **seq** qui permet de fabriquer des séquences. Ainsi la commande **s1 := seq(2 * i, i = 0..10)**; génère une séquence comportant les 11 premiers nombres pairs.

Remarque importante : si vous voulez que maple vous affiche de l'aide sur une commande, tapez **?commande**. Essayez par exemple **?seq**;

1. Obtenir une séquence s comportant les carrés des 20 premiers entiers.
2. A l'aide de seq, obtenir la séquence $\sin(x), \sin(2*x), \dots, \sin(15*x)$

Attention une fois la séquence obtenue on peut voir le i ème terme à l'aide des crochets mais on ne peut pas le modifier. Essayer les commandes **s[3]**; puis **s[3]:=12** par exemple.

1.2 Listes

Une autre structure très utile qui, elle, peut ensuite être modifiée est la **liste** (list pour maple). Une liste est tout simplement une séquence entourée de crochets. Par exemple **l:= [1,3,gkh]**; crée une liste; **l2:= [[1,3],[23,3],[-1,2]]** crée une liste de listes. Si s_1 est une séquence, **l:= [s1]** crée la liste formée des éléments de s_1 . Une opération utile est celle qui permet de faire le passage dans l'autre sens : si l est une liste, **op(l)** fournit la séquence correspondante. Une autre opération parfois utile est **nops(l)** qui fournit le nombre d'éléments de la liste.

1. A partir de s , créez une liste l , et transformez le troisième terme en 12. Puis obtenez la séquence correspondante (sans les [])

2. (optionnel). Une liste ressemble beaucoup à un tableau. maple connaît aussi les tableaux (array). On les utilisera moins. Pour maple si t fait référence à un tableau, t est en fait un pointeur. Comparez par exemple : **l1:= [1,4]; l2:= l1; l2[2]:= 3; l1;** et **t1:= array(1..2,[1,4]); t2:= t1; t2[2]:= 3; print(t1);** On peut transformer un tableau en liste à l'aide de la commande *convert* : essayer par exemple **l4:= convert(t1,list)**;

2 Manipulation de lois, utilisation de statevalf

On commencera systématiquement par charger les packages "stats", "statplots" et

“plots” en tapant *with(stats)*; puis *with(statplots)*: et enfin *with(plots)*:

Attention maple10 a un autre package très différent qui s'appelle *statistics*, mais il n'est pas disponible sous maple7.

En gros, il y a trois catégories de commandes spécifiques aux statistiques dont il faut se souvenir : les commandes précédées par *statevalf* permettent de faire des calculs sur des lois théoriques (obtenir la densité, la tracer, etc). Les commandes précédées par *describe* permettent de travailler sur des échantillons, et d'obtenir la moyenne empirique, l'écart-type empirique, etc. Enfin, la commande *random* permettra de générer un échantillon suivant une loi donnée.

On va travailler au début sur une variable aléatoire X supposée modéliser la taille d'une population, qui suit une loi normale $N(\mu, \sigma^2)$ avec $\mu = 175$ et $\sigma = 10$.

Pour raccourcir les commandes à taper, on va commencer par déclarer “ $X := \text{normald}[175, 10]$:”

Attention, pour maple, le deuxième paramètre est l'écart-type, pas la variance!

1. Obtenir la valeur de la densité en 180 en tapant *statevalf[pdf,X](180)*;
pdf est l'acronyme pour probability density function.
Obtenir également la valeur de la densité en 200.
Rappel : si vous voulez que maple vous affiche de l'aide sur une commande, tapez *?commande*. Essayez par exemple *?statevalf*. *?distributions* ou *?normald* vous fournit la syntaxe pour les différentes lois disponibles. *?stats* décrit tout le package. A utiliser sans modération...
2. *pdf* est donc utile pour les variables aléatoires à densité. Pour une variable aléatoire discrète, on peut obtenir la probabilité qu'elle vaille k à l'aide de *pf*; (pf pour probability function) par exemple, *statevalf[pf,binomiald[100,0.5]](35)*; fournit la probabilité qu'on obtienne 35 piles lors de 100 lancers successifs de pièces non pipées. Quelle est la probabilité pour que le lancer de 69 pièces pipées avec $P(\text{pile}) = 0.55$ fournisse 32 faces ?
3. Obtenir la valeur de la fonction de répartition de X en 180 avec *statevalf[cdf,X](180)*;
cdf est l'acronyme pour cumulative density function.
4. Quelle est la proportion des gens dans cette population dont la taille est comprise entre 170 et 180 ? Qui sont plus grands que 190 ?
5. Déterminer le quantile à 97,5% de X , c'est-à-dire le nombre q tel que $P(X \leq q) = 0.975$, en utilisant *statevalf* et l'inverse de la fonction de répartition (notée *icdf*).
6. Obtenir le graphe de la densité au moyen de *plot(statevalf[pdf,X],150..200)*;
(si f est une fonction, *plot(f(x),x=0..1)*; fournit le graphe de la fonction f entre 0 et 1)
7. Obtenir sur un même graphique la densité de X , et les droites verticales d'équation $x = 175$ et $x = q$. Pour tracer plusieurs courbes, on peut définir les tracés les uns après les autres, en leur donnant des noms, puis utiliser la commande *display*. Ainsi, par exemple,
P1 := plot(sin(x), x = 0..1) : P2 := plot(cos(x), x = 2..3) : display(P1, P2);
fournit un tracé avec le graphe du sinus et celui du cosinus entre 0 et 3.
Pour tracer une droite verticale, on peut utiliser la commande *plot([x(t),y(t),t = a..b])* qui trace une courbe paramétrée.
8. Obtenir une valeur approchée de l'espérance de X (et comparer à la vraie valeur...) en faisant par exemple
f1:=x->statevalf[pdf,X](x);
*evalf(int(x*f1(x),x=100..300))*; et comparer avec la vraie valeur.
La commande *x-> uneexpression* permet de définir une fonction (on pouvait aussi écrire *evalf(int(x*statevalf[pdf,X](x),x=100..300))*); *evalf* permet d'obtenir une valeur approchée, bien utile lorsque maple ne sait pas faire le calcul exact...
int(f(x),x=a..b) désigne $\int_a^b f(x)dx$.

9. Obtenir une valeur approchée de la variance.
10. Obtenir une valeur approchée du moment d'ordre 3 $E[X^3]$.

3 Statistiques sur un échantillon de taille n

En statistique, on ne connaît en général pas les détails de la loi qu'on observe, mais on a accès à des observations d'occurrence de cette loi, qu'on suppose indépendantes. On appelle cela un n -échantillon de loi X . Les statistiques sur les échantillons sont obtenues sous maple à l'aide de commandes de la forme *describe[quantité](echantillon)*, à condition que l'échantillon soit une liste.

1. Générer un 100-échantillon *Ech* de loi normale $N(175, 100)$ à l'aide de la commande *random Ech:=random[X](100)*;
random fournit une séquence. *Ech* obtenu ci-dessus est donc une **liste**, c'est-à-dire une suite de valeurs séparées par des virgules, et entourée par des crochets. On peut ne pas mettre les [] extérieurs mais dans ce cas à chaque fois qu'on manipule *Ech* dans la suite, il faut écrire [*Ech*].

2. Obtenir la moyenne empirique $\bar{x} = \frac{\sum_{i=1}^n X_i}{n}$ de l'échantillon de trois manières : en faisant une boucle pour calculer la somme, à l'aide de la commande **add** (utilisée pour sommer un nombre fixé de termes), et à l'aide de la commande *describe[mean](Ech)*;
Rappel : une boucle s'écrit **for i from 1 to 5000 do blabla end do**;
Faire **?add** pour connaître sa syntaxe.

3. Calculer l'écart-type empirique $\bar{\sigma} = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{x})^2}{n-1}}$ (attention au $n-1$...) de deux manières : à l'aide de *add*, et par *describe[standarddeviation[1]](Ech)*;

4. Calculer le quantile correspondant à 0.975 à l'aide de *describe[quantile[975/1000]](Ech)*, et comparer à la valeur théorique. Attention 0.975 ne marche pas, il faut vraiment une fraction.

5. Tracer un histogramme correspondant à l'échantillon *Ech* à l'aide de la commande **histogram** (Faites **?histogram** pour voir ses options : *numbars* permet d'imposer un certain nombre de classes, *area=1* permet d'être sûr que l'aire de l'histogramme est bien 1) Le représenter sur le même graphique que la vraie densité de X et que la densité estimée $N(\bar{x}, \bar{\sigma}^2)$.

6. La fonction de répartition empirique est la fonction qui à x associe la proportion des valeurs de l'échantillon inférieures ou égales à x . Pour tracer la fonction de répartition empirique F_{empi} on commence par trier l'échantillon à l'aide de *sort* (faire **?sort**). On obtient un échantillon trié *Echtri*. Que vaut $F_{empi}(Echtri[j])$ (Trouvez-le vous-mêmes, ne demandez pas à maple) ? Pour tracer la fonction de répartition empirique F_{empi} on va donc tracer les points de coordonnées $[Echtri[j], F_{empi}(Echtri[j])]$. Pour cela on utilise le principe suivant :

Pour tracer une suite de points de coordonnées $A[i], B[i]$ on peut utiliser **plot([seq([A[i], B[i]], i = 1..100)])**; **plot** prend donc ici en paramètre une liste de points, chaque point étant une liste de deux réels. Représenter sur le même graphique la fonction de répartition théorique et la fonction de répartition empirique. Il est habituel que l'accord soit bien meilleur pour la fonction de répartition qu'entre la densité et l'histogramme.

4 Challenge

J'ai mis sur ma page web (<http://www.math.u-bordeaux1.fr/~chabanol>), dans le répertoire Enseignement puis Statistiques, 6 échantillons obtenus en simulant des lois classiques. Récupérez-les

et faites-les lire par maple à l'aide de la commande *importdata*("toto"). Le nom du fichier doit être entre guillemets. Sous linux, l'adresse sera donnée sous la forme, par exemple, *importdata("/usr/local/Stat/TP/toto")*.

Sous windows, l'adresse peut être donnée de deux manières :

importdata("C : \\Stat\\TP\\toto") ou

importdata("C:/Stat/TP /toto").

Serez-vous capable de deviner quelle loi a été utilisée pour générer chaque échantillon ?