1 Commandes Xcas de proba-stats en 1ère et Terminale.

Ces commandes se trouvent dans le menu `Cmds->Proba-stats`.

Aléatoire	
`alea(n)`	renvoie un entier aléatoire entre 0 et $n-1$
`alea(a,b)`	renvoie un réel aléatoire entre a et b selon la loi uniforme
`randnorm(mu,sigma)`	renvoie un réel aléatoire selon la loi normale
`randexp(a)`	renvoie un réel aléatoire selon la loi exponentielle
`seq(expression,var,min,max)`	génère une séquence en remplaçant `var` dans `expression` par `min`, `min+1`, ... `max`
Statistiques	
`moyenne(liste)`	moyenne de la liste
`stddev(liste)` ou `stddevp`	écart-type divisé par n ou par $n-1$
`histogram(liste,min,size)`	histogramme d'une liste de valeurs, `min` (valeur minimale) et `size` (taille des classes) optionnels
Distributiuons	
`binomial(n,k)`	nombre de combinaisons de k parmi n
`binomial(n,k,p)`	probabilité d'obtenir k succès avec n tirages (où p est la probabilité de succès par tirage)
`binomial_cdf(n,p,n1,n2)`	probabilité d'obtenir entre $n1$ et $n2$ succès
`binomial_cdf(n,p,n1)`	probabilité d'obtenir au plus $n1$ succès
`binomial_icdf(n,p,t)`	renvoie le plus grand entier k tel que la probabilité d'obtenir au plus k succès soit $\leq t$
`normald(x)`	densité de la loi normale de moyenne 0 et écart-type 1
`normald(mu,sigma,x)`	densité de la loi normale de moyenne μ et écart-type σ
`normald_cdf(mu,sigma,x1)`	probabilité que x soit $\leq x1$
`normald_cdf(mu,sigma,x1,x2)`	probabilité que x soit entre $x1$ et $x2$ selon la loi normale de moyenne μ et écart-type σ
`normald_icdf(mu,sigma,t)`	renvoie x tel que la probabilité d'être $\leq x$ soit $\leq t$ (selon la loi normale de moyenne μ et écart-type σ)

N.B. : il peut être nécessaire d'utiliser la version 0.9.6 ou ultérieure de Xcas.

2 Exemples d'illustrations

– **Loi binomiale**
On tire une pièce à pile ou face 100 fois de suite. La probabilité d'obtenir 40 piles si la pièce est bien équilibrée est de `binomial(100,40,0.5)` (valeur approchée) ou `binomial(100,40,1/2)` (valeur exacte).
La probabilité que le nombre de piles soit inférieure ou égal à 40 vaut `binomial_cdf(100,0.5,40)` (attention l'ordre des paramètres est inversé par rapport à `binomial`).
On a une probabilité d'être dans l'intervalle de confiance $[50-\sqrt{100}, 50+\sqrt{100}]$ valant `binomial_cdf(100,0.5,40,60)` (ici environ 0.965).
Pour avoir un intervalle de confiance de 99% centré, on calcule `binomial_icdf(100,0.5,0.995)` qui renvoie 63 et `binomial_icdf(100,0.5,0.005)` qui renvoie 37 (symétrique de 63).
Si la pièce n'est pas équilibrée, remplacer 0.5 par la probabilité d'obtenir pile.

– **Simulation**
On peut simuler un pile ou face par `alea(2)`, une série de 100 par `seq(alea(2),j,1,100)`, sa somme par `sum(alea(2),j,1,100)` et l'histogramme de 1000 séries de 100 tirs par
`histogram(seq(sum(alea(2),j,1,100),k,1,1000));`
On peut superposer la loi binomiale au graphe en ajoutant à la fin de la commande précédente
`plotlist(seq(binomial(100,k,0.5),k,0,100),couleur=rouge)`
Pour une pièce non équilibrée, si p est la probabilité d'obtenir pile, on pourra utiliser le résultat du test `alea(0,1)<p` qui vaut 1 si le nombre généré est plus petit que p et 0 sinon et a donc une probabilité p. Par exemple `seq(alea(0,1)<p,j,1,100)`.
Pour simuler par exemple 100 erreurs selon la loi normale, faire
`seq(randnorm(0,1),j,1,100)`.
– **Convergence vers la loi normale**.
On peut tracer le diagramme en batons de la loi binomiale de paramètres par exemple $n = 100$ et $p = 0.4$ en tapant
`n:=100; p:=0.4; diagramme_batons(seq(binomial(n,k,p),k,0,n));`
Cliquer plusieurs fois sur le bouton bleu flèche vers le bas pour voir tout le diagramme. Pour superposer la loi normale qui approche ce diagramme, ajouter la commande
`graphe(normald(n*p,sqrt(n*p*(1-p)),x),x=0..100)`
Pour faire observer le théorème de Moivre-Laplace tel qu'il est énoncé, les commandes sont plus complexes. Il faut générer un histogramme avec en abscisse des intervalles de valeurs succesives (ici centrés) que prend $(X_n - np)/\sqrt{np(1-p)}$ et en ordonnée la loi binomiale, on tapera les commandes
```
S:=seq([(k-n*p)/sqrt(n*p*(1-p)),binomial(n,k,p)],k,0,n);
H:=histogram(S,affichage=nom_cache);
graphe(normald(x),x=-10..10,couleur=rouge)
```
On peut calculer l'aire d'une partie de l'histogramme avec la commande `aire(H[a..b-1])` (rectangles d'indices a à b) ou `binomial_cdf(n,p,a,b-1)`, que l'on peut comparer avec l'aire sous la courbe `int(normald(x),x=a..b)`. On peut reprendre les commandes précédentes en faisant varier n et p de manière interactive en définissant les paramètres n et p (menu `Edit->New parameter`, prendre garde de décocher la case `symb` pour avoir un curseur numérique et définir la plage de n et p ainsi que le pas).
– **Fluctuations à un seuil donné**
  – Visualisation de la partie hors de l'intervalle de confiance. Commencer par tracer la loi normale
  `graphe(normald(x),x=-10..10);`
  puis valider, puis ajouter à la fin de la commande
  `plotarea(normald(x),x=-6..-2); plotarea(normald(x),x=2..6)`
  Faire de même en remplaçant 2 et -2 par 3 et -3.
  – Calcul de l'intervalle de confiance centré au seuil de 1% pour la loi normale centrée réduite :
  `[normal_icdf(0,1,0.005),normal_icdf(0,1,0.995)]`