ECE 1 - Année 2015-2016
Lycée français de Vienne
Mathématiques - F. Gaunard
http://frederic.gaunard.com

# Des probabilités avec SciLab

*Séance de Jeudi 17 Décembre*

*La fonction* `rand` *renvoie un nombre au hasard selon une loi de probabilité définie à l'avance. Par exemple, en tapant* `rand('uniform')`*, on indique à la machine, que chaque appel* `rand()` *simule une loi uniforme* $\mathcal{U}([0;1])$.

**Exercice 1.** (Un dé à 6 faces)

(1) Écrire une instruction qui permet de simuler le jet d'un dé (non truqué) à six faces.
(2) Écrire ensuite un programme qui, à partir d'un nombre $n$ de lancers entré par l'utilisateur renvoie la fréquence d'obtention d'un 6.
(3) Faire tourner le programme précédent pour des valeurs de plus en plus grandes de $n$. Que semble-t-on observer?

On utilise dans ce premier exercice la fonction `rand()` (chargée avec la loi uniforme) pour **simuler** le lancer d'un dé. Il faut cependant ajuster un peu les choses: la fonction `rand()` renvoie un nombre aléatoire (réel donc éventuellement décimal) entre 0 et 1. Or ici, on veut u nombre entier entre 1 et 6. L'idée est donc de multiplier le résultat par 6 puis de prendre **la partie entière** et de rajouter 1. Le résultat sera donc bien un nombre entier entre 1 et 6. Enfin, si l'on veut écrire une fonction qui renvoie, pour un argument de $n$ lancers, la fréquence des 6 obtenue, il suffit d'intégrer une boucle `for` et de compter (à l'aide d'un test d'égalité) le nombre de 6 obtenus puis de diviser par $n$. On constate que, pour un grand nombre de lancers simulés, la fréquence observée se rapproche de la probabilité théorique d'obtention d'un 6, à savoir 1/6.

```
function y=frequence6(n)
    f=0;
    for i=1:n
        if floor(6*rand())+1==6 then
            f=f+1;
        end
    end
    y=f/n;
endfunction
```

**Exercice 2.** (Un jeu et un compteur)

Écrire un programme qui, après avoir tiré (secrètement) au hasard un nombre entier entre 1 et 10, demande à l'utilisateur de le deviner et compte le nombre de coups nécessaires.

On utilise la même idée que précédemment pour "transformer" un nombre aléatoire (réel) entre 0 et 1 en nombre aléatoire entier compris entre 1 et 10 (on pourra d'ailleurs s'amuser à rendre le jeu plus difficile en augmentant les possibilités mais en donnant des indications à chaque mauvaise réponse). On programme le jeu comme suit:

```
r=floor(10*rand())+1; //choix du nombre aléatoire
x=input("Essayer de deviner le nombre mystère :");
n=1; //initialisation du nombre de tentatives
while x<>r
    x=input("Mauvaise réponse. Essayez encore :");
    n=n+1; //une tentative de plus
end
disp("Bravo! Vous avez trouvé le nombre mystère en "+string(n)+" coups.")
```

On a choisi d'afficher le résultat en une seule phrase. Pour cela, on a du transformer le contenu de la variables $n$ (le nombre de coups) en une chaîne de caractères à l'aide de la fonction `string()`. On "recolle" tous les morceaux en une seule chaîne de caractères avec des `+`.

On vient de programmer un premier petit jeu dont on peut être assez fier et qu'on va tester tout de suite. Bon étrangement, on s'en lasse assez vite....

```
Essayer de deviner le nombre mystère :3
Mauvaise réponse. Essayez encore :2
Mauvaise réponse. Essayez encore :1

 Bravo! Vous avez trouvé le nombre mystère en 3 coups.
```