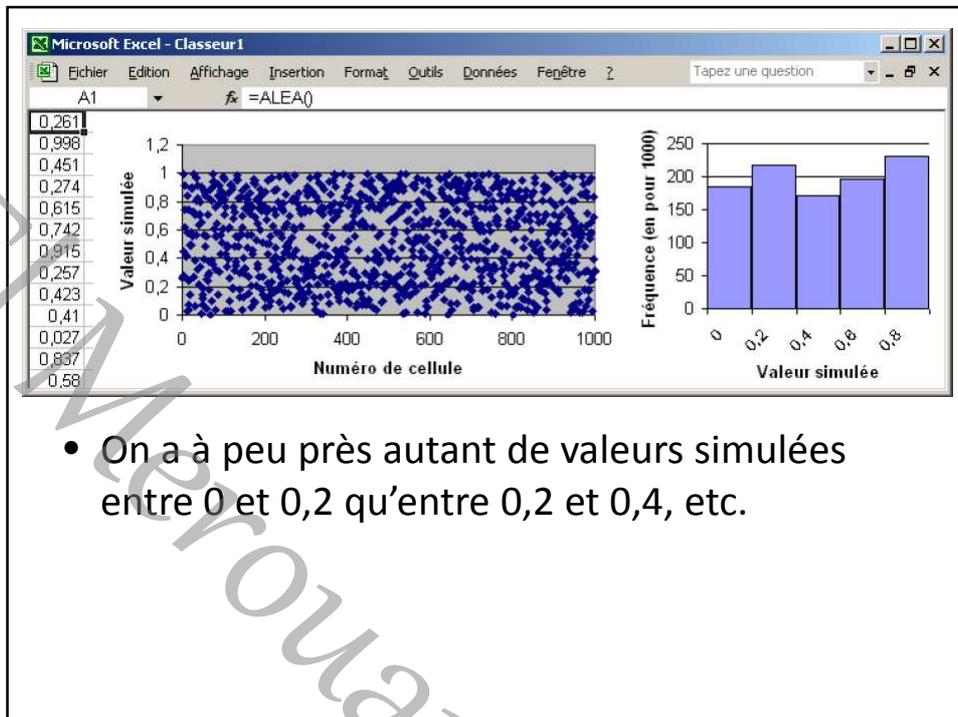


Générations des nombres aléatoires (2)

Prof. Mohamed El Merouani

Génération des nombres pseudo aléatoires avec Excel

- Nous pouvons simuler la loi « uniforme » avec Excel à partir de la fonction **ALEA**.
- Pour le vérifiez, recopions la formule **=ALEA()** sur **1000 cellules** et représentons la **distribution des valeurs obtenues** par un nuage de points ou par un histogramme.



- Nous aurions pu produire une série équivalente en passant par l'utilitaire **Génération de nombres aléatoires (Allez dans le menu Outils. Si l'Utilitaire d'analyse n'apparaît pas, installez le en passant par Macros complémentaires).**



Génération de nombres aléatoires

Nombre de variables: 1 OK

Nombre d'échantillons générés: 1000 Annuler

Distribution: Uniforme Aide

Paramètres

Entre 0 et 1

Entier générateur: 123456

Options de sortie

Plage de sortie:

Insérer une nouvelle feuille:

Créer un nouveau classeur

- Par rapport à la fonction **ALEA**, dont le résultat est volatile, l'utilitaire permet de reproduire la même série. Il suffit d'entrer le même Entier générateur (ici 123456).

GÉNÉRATEURS DE NOMBRE ALÉATOIRE sous R

- Pour générer « n » nombres aléatoires qui suivent une loi uniforme (0,1) (comme nous l'avons vu en Excel) la commande en R est:
> runif(n, min=0, max=1)
- Si, nous voulons les représenter sur un graphique, on écrit:
>x=runif(n, min=0, max=1)
- Alors R les garde le résultat dans x sans l'afficher. Puis on trace le nuage de points:
>plot(x)
- Si on tape x tout simplement, R les affiche.

GÉNÉRATEURS DE NOMBRE ALÉATOIRE sous R

- Pour la reproductibilité et le contrôle, la graine pour les générateurs de nombre aléatoire devrait être mise explicitement, avant la première utilisation d'un générateur de nombre aléatoire dans le Script.
- La commande est :
`>set.seed(m)`
- En changeant la graine et en lançant de nouveau le Script, on obtiendra de différents (indépendants) échantillons à partir des générateurs de nombre aléatoire.

Générateur de nombres pseudo-aléatoires en Scilab:

- Scilab possède une fonction `rand()` dont les appels successifs fournissent une suite de variables aléatoires indépendantes et identiquement distribuées, de loi uniforme sur $[0, 1]$.
- En fonction des arguments d'entrée, la fonction `rand()` peut renvoyer une matrice de doubles aléatoires ou peut configurer ou récupérer la distribution des nombres aléatoires ou peut configurer ou récupérer la graine du générateur aléatoire.

Générateur de nombres pseudo-aléatoires en Scilab:

- `r=rand()` //renvoi un scalaire aléatoire.
- `r=rand(m1,m2)` //est une matrice aléatoire de dimensions $m1 \times m2$.
- `r=rand(m1,m2,..,mn)` //est une matrice aléatoire de dimensions $m1 \times m2,.. \times mn$.
- `r=rand(a)` //est une matrice aléatoire de mêmes dimensions que `a`. `rand(a)` est complexe si `a` est une matrice complexe.

Changer la distribution des nombres aléatoires:

- `rand("uniform")` //La loi par défaut est uniforme sur l'intervalle $[0,1]$.
- `rand("normal")` //La loi par défaut est une loi normale centrée réduite (moyenne 0, variance 1).
- `key=rand("info")` //renvoie la distribution courante, c'est à dire "uniform" ou "normal".
- `key` est un argument (chaîne de caractères, "uniform" ou "normal "). Par défaut: `key="uniform"`

Exemple:

Dans cet exemple, nous générons 1000 nombres aléatoires en une seule colonne qui suivent la loi uniforme (0,1) puis nous allons tracer l'histogramme correspondant en lui ajoutant le titre et le noms des axes.

```
r=rand(1000,1,"uniform");  
scf\(\);  
histplot(10,r);  
xtitle("Nombres uniformes par rand","X","Fréquence")
```

- http://help.scilab.org/docs/5.4.0/fr_FR/rand.html