ECE 2 - Année 2021-2022
ENC Bessières
Mathématiques - F. Gaunard
http://frederic.gaunard.com

T.P. n°1

Simulations de variables aléatoires (finies et) discrètes
Statistiques descriptives univariées

1 Simulation de v.a.r suivant une loi discrète à l'aide de la fonction `rand()`

1.1 Simulation d'une v.a.r suivant une loi de Bernoulli

(1) Rappeler la définition, la loi et les résultats concernant la loi de Bernoulli.

(2) Dessiner, à la main, la distribution théorique des valeurs d'une variable de Bernoulli.

(3) On considère la fonction ci-dessous, que l'on recopiera et enregistrera sous le nom `bernouilli.sci`. Que fait-elle ?

```
function y = bernoulli (p)
          r=rand();
          if r < p then
               y = 1;
          else
               y = 0;
          end
endfunction
```

(4) Écrire une fonction `v=SampleBernoulli(N,p)` qui prend en paramètre un entier `N` non nul, un réel `p` de $]0;1[$ et renvoie un vecteur ligne `v` contenant le résultat de la simulation de N v.a.r indépendantes de loi $\mathcal{B}(p)$.

(5) Évaluer la commande `U=SampleBernoulli(1000, 0.3); Y=tabul(U, "i")`. On rappellera notamment ce que fait la commande `tabul()`.

(6) Que font les instructions suivantes? Interpréter.

```
m=mean(U);
v=mean(U-m).^2;
```

(7) Taper ensuite `clf(); bar(Y( : , 1), Y( : , 2))`. Expliquer le fonctionnement de `bar` en détaillant le contenu de `Y( : , 1)` et `Y( : , 2)`.

(8) Comparer le résultat obtenu avec la réponse à la question (2). Est-ce cohérent ?

(9) Le diagramme obtenu représente l'effectif de chaque classe. Comment modifier l'appel pour obtenir un diagramme des fréquences?

1.2 Simulation d'une v.a.r suivant une loi binomiale

(1) Rappeler la définition, la loi et les résultats concernant la loi binomiale.

(2) La fonction suivante, permet de représenter, sous `SciLab`, la distribution théorique des valeurs d'une variable de loi binomaile de paramètre n et p.

```
function [ ]=dist_th_bin(n,p)
    P=zeros(1,n+1)
    for k=1:n+1
        P(k)=prod(n+2-k:n)/prod(1:k-1)*p^(k-1)*(1-p)^(n+1-k)      //P(k)
correspond à P(X=k-1)
    end
    bar(0:n, P, width=0.2)
endfunction
```

Recopier et exécuter ce programme pour différentes valeurs de n et p.

(3) Écrire une fonction `y=binomiale(n,p)` qui prend en argument un entier non nul `n`, un réel `p` de $]0;1[$ et renvoie une variable y résultat de la simulation d'une v.a.r de loi binomiale $\mathcal{B}(n,p)$.

(4) Écrire une fonction `v=SampleBinomiale(N,n,p)` qui prend en argument deux entiers non nuls `N` et `n`, un réel `p` entre 0 et 1 et renvoie un vecteur ligne `v` contenant le résultat de la simulation de `N` v.a.r indépendantes de loi $\mathcal{B}(n,p)$.

(5) Évaluer `U=SampleBionmiale(1000, 10, 0.3)`. Comparer la moyenne et la variance empiriques aux valeurs théoriques. Calculer de deux façons l'écart-type. Détailler les commandes permettant d'obtenir le tracé d'un diagramme des effectifs à l'aide de la fonction `bar`.

(6) Comment peut-on obtenir le même résultat en faisant appel à la fonction `histplot`?

(7) Que font les commandes suivantes?

```
m=tabul(U, "i");
effc=cumsum(m(:,2) );
frec=effc/sum(m(:, 2));
```

Remarque 1. La fonction `histplot` convient davantage à des v.a.r à densité. Par exemple, l'appel `histplot(0:n, sampleBinomiale(N, n, p))` ne convient pas : `Scilab` considère alors que la première classe est $[0;1]$ (et non pas $[0;1[$!).

1.3 Simulation d'une v.a.r suivant une loi géométrique

(1) Rappeler la définition, la loi et les résultats concernant la loi géométrique.

(2) Écrire une fonction `SciLab` permettant de représenter la distribution théorique des 20 premières valeurs d'une variable de loi géométrique.

(3) On considère la fonction ci-dessous, que l'on recopiera et enregistrera sous le nom `geometrique.sci`. Que fait-elle ? Est-on sûr que cette fonction s'arrête toujours?

```
function y = geometrique (p)
          r=bernoulli(p);
          y = 1;
          while r >= p
              y = y+1;
              r=bernoulli(p);
          end
endfunction
```

(4) Écrire une fonction `v=SampleGeometrique(N,p)` qui prend en paramètre un entier `N` non nul, un réel `p` de $]0;1[$ et renvoie un vecteur ligne `v` contenant le résultat de la simulation de `N` v.A.R indépendantes de loi $\mathcal{G}(p)$.
(5) Évaluer `U=SampleGeometrique(1000, 0.4)`. Comparer les moyenne et variance empiriques aux valeurs théoriques.
(6) Détailler les commandes permettant d'obtenir le tracé d'un diagramme des effectifs à l'aide de la fonction `bar`.
(7) Peut-on obtenir le même résultat en utilisant la fonction `histplot`?

2 Simulation de v.a.r suivant une loi discrète à laide de la fonction `grand()`

☞ Compléter le texte ci-dessous

- `grand(1,N,      ,a,b)` génère 1 échantillon de N nombres aléatoires suivant la loi $\mathcal{U}([\![a;b]\!])$.
- `grand(              )` génère m échantillons de N nombres aléatoires suivant la loi $\mathcal{U}([\![a;b]\!])$.

- `grand(1,N,"bin",n,p)` génère 1 échantillon de N nombres aléatoires suivant la loi   .
- `grand(              )` génère m échantillon de N nombres aléatoires suivant la loi   .

- `grand(1,N,"geom",n,p)` génère 1 échantillon de N nombres aléatoires suivant la loi   .
- `grand(              )` génère m échantillon de N nombres aléatoires suivant la loi   .

- `grand(1,N,      ,lam)` génère 1 échantillon de N nombres aléatoires suivant la loi $\mathcal{P}$(`lam`).
- `grand(              )` génère m échantillons de N nombres aléatoires suivant la loi $\mathcal{P}$(`lam`).

2.1 Comparaisons de distributions

(1) Écrire un programme (et l'enregistrer sous le nom `distrib_geom.sce`) qui
- demande à l'utilisateur d'entrer la valeur d'un paramètre `p`:
- demande à l'utilisateur d'entrer la valeur d'un paramètre `N`;
- réalise la simulation d'un échantillon de `N` v.a.r. indépendantes de loi géométrique $\mathcal{G}$(`p`) (on stocke le résultat obtenu dans une variable `obs`);
- trace le diagramme des effectifs de cette simulation.

(2) Ajouter `width=0.4,'red'` dans l'appel à la fonction `bar`. À quoi servent ces arguments optionnels?

(3) Comment modifier le programme précédent afin d'obtenir le diagramme des fréquences ? Le faire.

(4) On ajoute les lignes suivantes

```
x = 1:10
y = (1-p) ^ (x-1) * p
bar(x, y, width=0.4)
```

(a) Que contient la variable `y`? Que cela représente-t-il pour une v.a.r X telle que $X \hookrightarrow \mathcal{G}(p)$?
(b) Représenter simultanément les diagrammes à bâtons des valeurs théoriques et empiriques. On ajustera le programme afin d'éviter toute superposition des bâtons.

(5) Écrire un programme `distrib_poisson.sce` analogue au précédent pour la loi de Poisson et comparer une fois encore les diagrammes des valeurs théoriques et empiriques.
☞ Afin de calculer les valeurs théoriques de la loi de Poisson, on pourra écrire sa propre fonction `facto()` pour calculer $k!$.

3 Autres exercices

Exercice 1. (D'après **ECRICOME 2017**) Soit n un entier naturel non nul.
On effectue une série illimité de tirages d'une boule avec remise dans une urne contenant n boules numérotées de 1 à n. On considère la variable aléatoire T_n égale au nombre de tirages nécessaires pour que, pour la première fois, la somme des numéros des boules obtenues soit supérieure ou égale à n. Soit également Y une variable aléatoire à valeurs dans $\mathbb{N}^*$ telle que:

$$\forall k \in \mathbb{N}^*,\ P(Y=k) = \frac{k-1}{k!}.$$

(1) Écrire une fonction `y=T(n)` permettant de simuler la variable T_n.

On peut montrer (et en fait le sujet le demandait) que (T_n) *converge en loi* vers T, c'est à dire que, tout entier naturel k non nul,

$$\lim_{n\to+\infty} P(T_n = k) = P(Y=k).$$

(2) Recopier et exécuter le script suivant pour les valeur de n suivantes:

$$n=5, \qquad n=10, \qquad n=50, \qquad n=100, \qquad n=1000.$$

```
function y=freqT(n)
    y=zeros(1,n);
    for i=1:100000
        k=T(n);
        y(k)=y(k)+1;
    end
    y=y/100000;
endfunction

function y=loitheoY(n)
    y=zeros(1,n);
    for i=1:n
        y(k)=(k-1)/prod(1:k);
    end
endfunction

n=input('n=?');
plot2d(loitheoY(6), -2)
x=freqT(n)
bar(x(1:5), .05)
```

(3) Expliquer ce que représentent les vecteurs renvoyés par les fonctions `freqT` et `loitheoY`. Comment ces vecteurs sont-ils représentés graphiquement dans chaque graphique obtenu ?

(4) Expliquer en quoi cette succession de graphiques permet d'illustrer le résultat susnommé?

Exercice 2. (D'après **ECRICOME 2015**)

On effectue des tirages **sans remise** une urne U contenant $(N-1)$ boules blanches $(N \geq 3)$ et une boule noire, jusqu'à l'obtention de la boule noire.

On note X la variable aléatoire qui prend pour valeur le nombre de tirages nécessaires pour l'obtention de la boule noire.
On notera pour tout entier naturel i non nul:

- N_i l'événement "on tire une boule noire lors du i-ième tirage" .
- B_i l'événement "on tire une boule blanche lors du i-ième tirage".

(1) On simule 10000 fois cette expérience aléatoire.
Recopier et compléter le programme `SciLab` suivant pour qu'il affiche l'histogramme donnant la fréquence d'apparition du rang d'obtention de la boule noire:

```
N=input('N=?');
S=zeros(1, N);
for k= 1 : 10000
    i=1;
    M=N;
    while ................................
        i=i+1;
        M=................................
    end
    S(i)=S(i)+1;
end
disp(S / 10000)
bar(S / 10000)
```

(2) Exécuter le programme complété ci-dessus en entrant 5 au clavier. Quelle conjecture pouvez-vous émettre sur la loi de la variable aléatoire X à partir de l'histogramme obtenu?

Exercice 3. (D'après **EDHEC 2004**)
On désigne par n un entier naturel supérieur ou égal à 2. On lance n fois une pièce équilibrée (c'est-à-dire donnant *pile* avec la probabilité 1/2 et *face* également avec la probabilité 1/2), les lancers étant supposés indépendants.
On note Z la variable aléatoire qui vaut 0 si l'on n'obtient aucun pile pendant ces n lancers et qui, dans le cas contraire, prend pour valeur le rang du premier pile.
Recopier et compléter le programme suivant pour qu'il renvoie une simulation de la variable aléatoire Z.

```
function y=EDHEC2004(n)
    y=0;
    i=1;
    while y==0 & i<n
        r=................................
        if ......................... then
            y=.............................
        end
        i=i+1;
    end
endfunction
```

Exercice 4. (D'après **EML 2018**)

Soit p un réel de $]0;1[$. Deux individus A et B s'affrontent dans un jeu de *Pile* ou *Face* dont les règles sont les suivantes:

- le joueur A dispose d'une pièce amenant *Pile* avec la probabilité $\dfrac{2}{3}$ et lance cette pièce jusqu'à l'obtention du deuxième *Pile* ; on note X la variable aléatoire prenant la valeur du nombre de *Face* alors obtenus ;
- le joueur B dispose d'une autre pièce amenant *Pile* avec la probabilité p et lance cette pièce jusqu'à l'obtention d'un *Pile* ; on note Y la variable aléatoire prenant la valeur du nombre de *Face* alors obtenus ;

- Le joueur A gagne si son nombre de *Face* obtenus est inférieur ou égal à celui de B ; sinon c'est le joueur B qui gagne.

On dit que le jeu est équilibré lorsque les joueurs A et B ont la même probabilité de gagner.

(1) Écrire une fonction `SciLab` d'en-tête `function x = simule_X()` qui simule la variable aléatoire X.
(2) On suppose que l'on dispose d'une fonction `simule_Y` qui, prenant en argument un réel p de $]0;1[$, simule la variable aléatoire Y. Expliquer ce que renvoie la fonction suivante :

```
function r = mystere(p)
    r = 0
    N = 10^4
    for k = 1:N
        x = simule_X()
        y = simule_Y(p)
        if x <= y then
            r = r + 1/N
        end
    end
endfunction
```

(3) Tracer, en fonction de p, une estimation de la probabilité que A gagne et conjecturer, à la vue du graphique obtenu, une valeur de p pour lequel le jeu serait équilibré.

Exercice 5. On effectue une succession infinie de lancers indépendants d'une pièce donnant pile avec probabilité p. On note $q = 1 - p$.
On dit que la première série est de longueur $n \geq 1$ si les n premiers lancers ont amené le même côté de la pièce et le $(n+1)$-ième l'autre côté. De même la deuxième série commence au lancer suivant la fin de la première série et se termine au lancer précédant un changement de côté.

Soient X_1 et X_2 les variables aléatoires égales aux longueurs de la première et deuxième série.

(1) Recopier et compléter la fonction suivante, prenant en argument la probabilité p d'obtenir pile et permettant de simuler la variable aléatoire X_1.

```
function X=longueur(p)
    X=1;
    told=grand(1,1,'bin', ....., .....)
    tnew=grand(1,1,'bin', ....., .....)
    while .............
        X=.................
        told=tnew;
        tnew=..............
    end
endfunction
```

(2) Écrire une fonction d'entête `function U=SampleX1(N,p)` permettant d'obtenir un $N-$échantillon de X_1, c'est à dire un vecteur ligne U de taille N dont chaque composante est une réalisation de la variable X_1.

(3) Écrire un script permettant de comparer graphiquement les fréquences des valeurs obtenues sur un échantillon de taille 1000 de X_1 et les valeurs théoriques de la loi géométrique de paramètre $1/2$. Commenter.