ENSIAS

Rapport de Projet de Fin de Première Année

Etudes de générateurs de loi

Réalisé par : Nossair Errahmani & Meryam Daoudi

Année Scolaire: 2017/2018

Remerciements:

Nous tenons à remercier Mr. Amrani pour ses efforts considérables au long du semestre pour nous avoir accordé de son temps, orienté nos recherches, et soutenu dans notre travail.

Nous tenons également à remercier corps enseignant de l'ENSIAS pour nous avoir tendu une main salvatrice et inestimable toute l'année, et de nous avoir inculqué les fondations de tout ce dont nous avons eu, avons et auront besoin cette année comme plus tard dans notre parcours, nous leur exprimons notre sincère gratitude.

Nous souhaitons enfin remercier les membres du jury pour nous accorder de leur temps précieux pour nous donner la chance de présenter le fruit de notre travail acharné.

Table des Matières:

Résumé:

Notre but est de comparer les générateurs utilisés par les logiciels étudiés (Matlab, R, Excel, SPSS, Linéaire, Hamlili) à la lumière de tests qui visent à quantifier leur qualité, en l'occurrence le test Khi-2 et le test Kolmogorov-Smirnov.

L'un cherche à savoir si un échantillon d'une loi discrète suit bien la répartition à laquelle on s'attendrait en théorie pour un échantillon de cette taille, l'autre cherche à savoir si un échantillon d'une loi continue est assez conforme pour être utilisé comme approximation de cette dernière.

Ensuite, il est pertinent de se demander s'il existe une corrélation entre les résultats des deux tests sur un même échantillon qui pour l'un reste intact et pour l'autre subit une transformation inverse.

Enfin, on cherchera à savoir quelle propriété rentre en jeu pour déterminer la valeur de K dans le test KS, et on cherchera à établir un algorithme qui permet de relier les valeurs de cette propriété, nommément la discrépance, à la valeur plus ou moins bonne de K.

Mots-Clés: Test d'ajustement Khi-2, test de Kolmogorov-Smirnov, génération de nombres aléatoires, discrépance

Abstract:

Our goal is to compare the generators used by the softwares we're focusing on (Matlab, R, Excel, SPSS, Linear, Hamlili) using tests that aim to translate their quality into a grade, namely Chi-2 and Kolmogorov-Smirnov.

The former looks to show that a discrete law's sample follows the distribution we expect for a sample of that size. The latter makes sure that a continuous law's sample is conform enough to use it as an approximation for it.

Then it is relevant to wonder if there exists a correlation between the results of both tests on the same sample, who stays the same for one and undergoes a reverse transformation for the other.

Finally, we'll try to find what kind of property comes into play to determine K's value, and we'll try to establish an algorithm that allows us to link the values of that property, namely discrepancy, and the more or less satisfying values of K..

Keywords: Chi-2 Adjustment test, Kolmogorov Smirnov test, random number generation, discrepancy

Table des figures:

Introduction Générale:

Lorsqu'on cherche à générer du hasard, il faut bien choisir ses méthodes. Il existe une quête à la recherche d'une source de hasard absolu, chose à laquelle on aspire au moins à s'approcher le plus possible tant elle parait impossible à atteindre.

Pour cela, plusieurs techniques sont disponibles, et des générateurs incorporés dans des logiciels bien connus en ont chacun choisi une, avec plus ou moins de succès. Le bon choix de ces propriétés génératrices sont absolument primordiales pour garantir un pseudo-hasard convaincant, tant les enjeux sont grands.

C'est pour cela que, devant une tâche requérant une génération de nombres aléatoires, la première question à se demander est 'Quel générateur doit-on utiliser ?', et c'est à cette question que nous tenterons de répondre dans ce mémoire.

Chapitre 1:

Lois Discrètes et test Khi-2

1. Lois discrètes et test Khi-2:

1.1. Forme générale de la méthode de transformation inverse :

Soit X une variable aléatoire discrète de fonction de répartition F.

Posons $X(\Omega) = \{x_1, x_2, \ldots, x_n\}$, $P(X=x_i) = p_i$, $\sum_{i=1}^{n} P(X = x_i) = 1$

On dispose d'une séquence u_i $i = 1,\ldots,m$ d'une loi uniforme $U_{[0, 1]}$.

La méthode de transformation inverse consiste à construire la variable « $\tilde{X}$ » tel que

$$\tilde{X} = \begin{cases} x_1 & \text{si} \quad U \leq F(x_1) \\ x_2 & \text{si } F(x_1) < U \leq F(x_2) \\ & \vdots \\ x_j & \text{si } F(x_{j-1}) < U \leq F(x_j) \\ & \vdots \\ x_n & \text{si } F(x_{n-1}) < U \leq F(x_n) \end{cases}$$

1.2. Génération de variable aléatoire X suivant la loi Bernoulli de paramètre p :

Soit X une variable aléatoire suivant la loi Bernouilli de paramètre p

$(p = P(X = 1) = 1 - P(X = 0))$.

On tire alors une variable $U \sim U([0, 1])$:

si $U < 1 - p$ on retient $X = 0$;

sinon on prend $X = 1$.

1.3. Test d'ajustement de Khi-2 :

Soit les hypothèses suivantes :

H_0 : La population suit la distribution «X»
H_1 : la population ne suit pas la distribution «X».

L'idée est de découper le domaine de la distribution en intervalles. Dans chaque intervalle, on calcule à partir de la loi spécifiée sous H_0 la fréquence théorique attendue. On compte ensuite combien d'observations l'on retrouve dans chaque intervalle. Il suffit alors de comparer les fréquences observées aux fréquences théoriques. Supposons que l'on divise la distribution en « k » intervalles.

Soit un intervalle « i » donné. La fréquence théorique attendue pour l'intervalle « i » est

$E_{i=np_i}$

Ainsi la statistique Khi-2 est :

$$Q = \sum_{1}^{k} \frac{(O_i - E_i)^2}{E_i} = \chi^2_{k-1}$$

Où Oi est la fréquence observée.

1.4. La procédure du test khi-2 :

La démarche à suivre :

- Formuler H_o (dans notre cas : la distribution observée n'est pas différente de la distribution supposée d'après la loi que l'on souhaite tester).
- Répartir les données en classes.
- déterminer le nombre de degrés de liberté à partir du nombre de classes.
- fixer un seuil de signification (la valeur 5 % est choisie par défaut dans le reste de notre étude).
- calculer algébriquement la distance entre les ensembles d'informations à comparer.
- déterminer Khi-2 théorique (déduire la distance critique à l'aide d'une table de χ^2 $P(K > 0.5) = \alpha(= 0.5)$).
- conclure si cette distance est supérieure à la distance critique (on en conclut ainsi que l'échantillon est rejeté ou que l'échantillon est valide).

1.5. Applications :

1.5.1. Test khi-2 pour le générateur Matlab :

À propos du générateur matlab:

Matlab utilise plusieurs algorithmes pour différentes fonctions.

Pour générer la loi uniforme, il utilise un générateur multiple (Matlab 4), une version modifiée de la soustraction de Marsaglia (5 à 7.3) et Mersenne Twister (à partir de 7.4).

Pour la loi normale, il utilise l'algorithme polaire (Matlab 4) et l'algorithme ziggurat de Marsaglia (à partir de 5).

Pour évaluer la qualité du générateur de la loi uniforme de Matlab, on applique le test de Khi-sur 100 échantillon de 10000 VAR suivant la loi Bernouilli (0.7) tout en utilisant la méthode des transformations inverses expliquée plus haut.

```
%méthode de transformation inverse
for j=1:100
    nb=[0,0];
    for i=1:10000
        u=rand;
        %génèration du nombre aléatoire par la fct rand de Matlab
        if u<0.3
            Val_ber(j,i)=0;
            nb(1)=nb(1)+1;
        end
        if u>=0.3
            Val_ber(j,i)=1;
            nb(2)=nb(2)+1;
        end
    end
    %calcul de la statistique pour chaque échantillon
    s(j)=((nb(1)-0.3*10000)^2)/(10000*0.3)+((nb(2)-0.7*10000)^2)/(10000*0.7);
end
```

Figure 1.a : Script utilisé pour la transformée inverse du gén. Matlab en loi Bernoulli

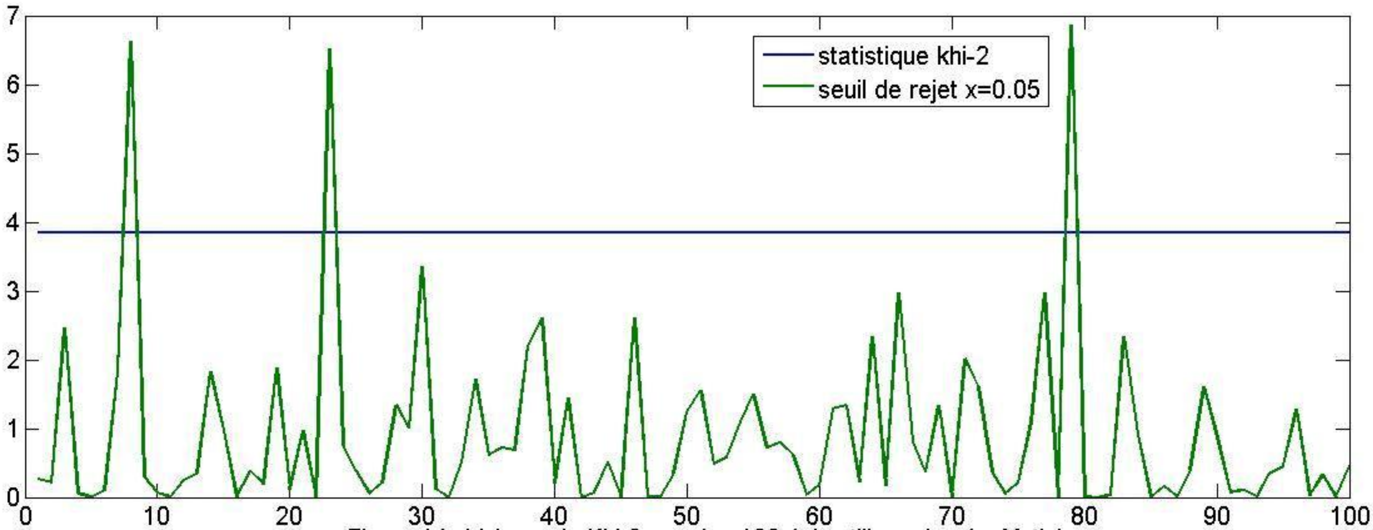


Figure 1.b: Valeurs de Khi-2 pour les 100 échantillons du gén. Matlab

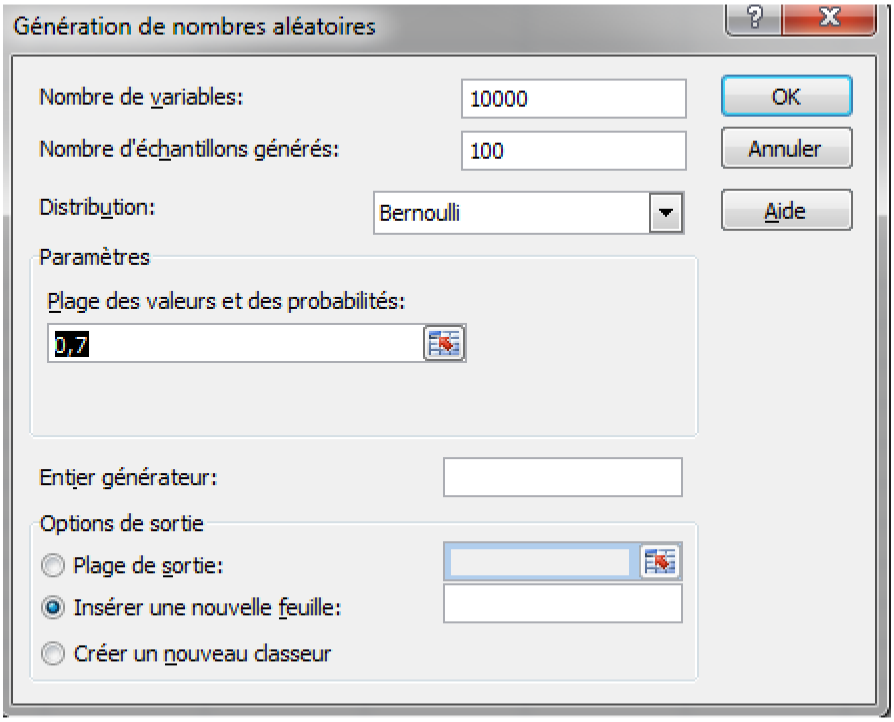
Ainsi, le générateur de Matlab présente 3 échantillons refusés, et 6 échantillons tangents, le reste étant valide.

1.5.2. Test Khi-2 pour le générateur Excel :

A propos du générateur Excel :

La fonction RAND dans les premières versions d'Excel utilisait un générateur de nombre pseudo aléatoire dont la performance de l'algorithme dans les tests standards d'aléatoires n'était pas suffisante. Malgré le fait que cela n'affectait que les utilisateurs qui appelleront la fonction plus d'un million de fois et donc par occurrence ne concernait pas la majorité des utilisateurs, une nouvelle version du générateur pseudo-aléatoire a été implémenté à partir d'Excel 2003, cette version ayant réussi toute la batterie des tests que l'ancien n'a pas pu faire. (test de Diehard).

Ainsi, on applique le test Khi-2 sur un échantillon de variables aléatoires suivant la loi Bernoulli(0.7) généré par l'utilitaire d'analyse d'Excel.



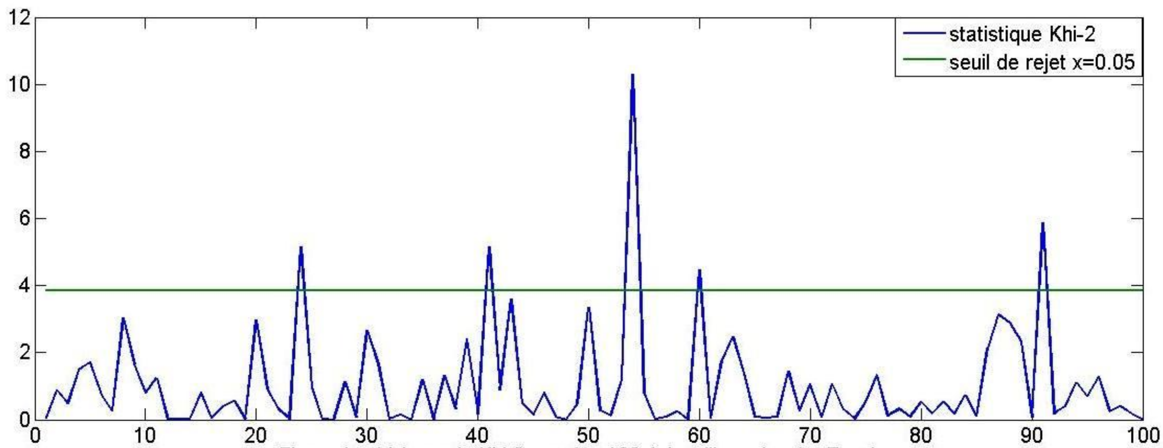


Figure 1.c: Valeurs de Khi-2 pour les 100 échantillons du gén. Excel

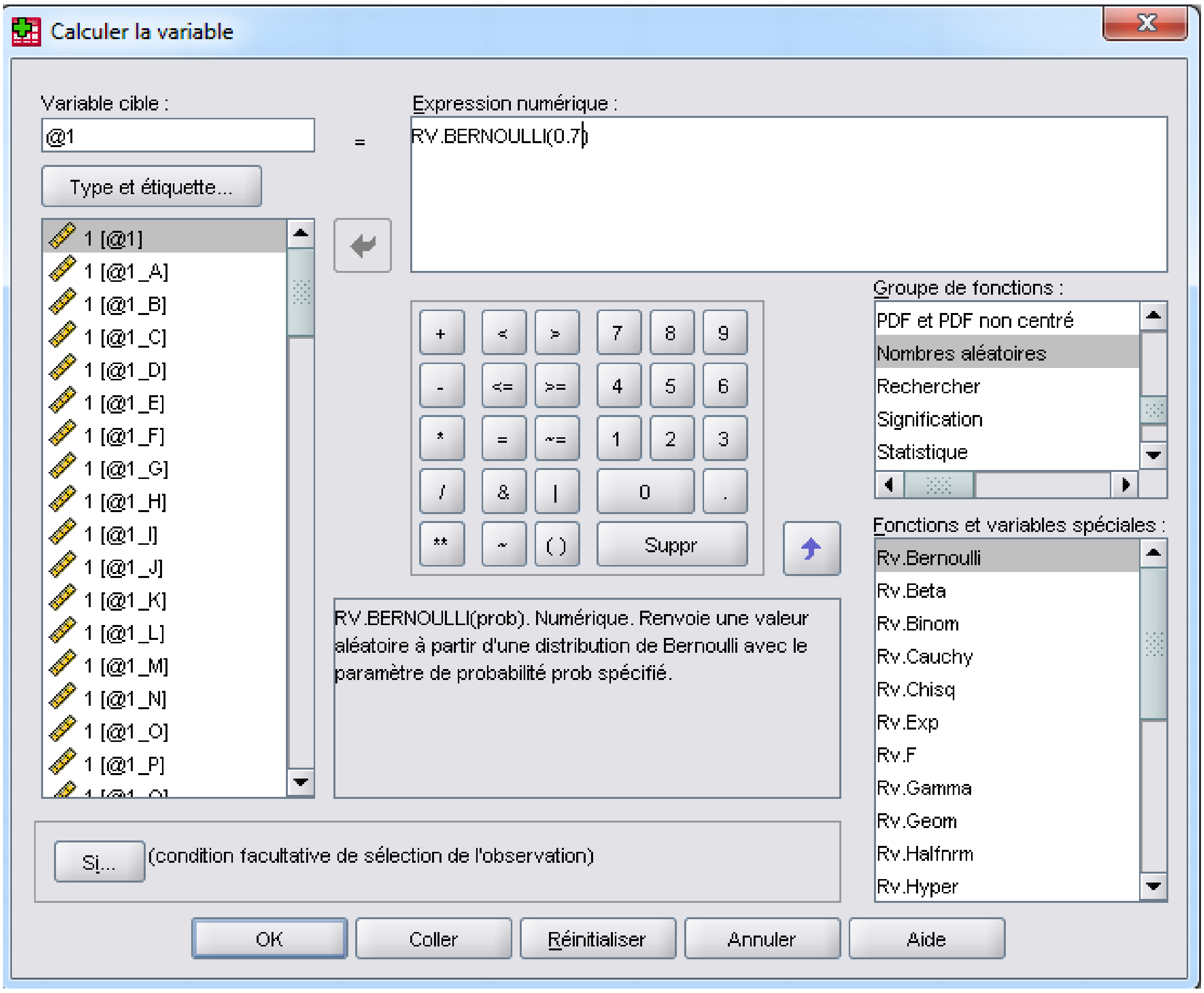
Donc d'après les résultats du test khi-2, le générateur Excel présente 5 échantillons rejetés, et 9 échantillons tangents.

1.5.3. Test Khi-2 pour le générateur SPSS:

À propos du générateur :

Jusqu'à la version 12.0, SPSS utilisait un algorithme recommandé par Fishman&Moore dans leur article paru en 1981: "In Search of Correlation in Multiplicative Congruential Generators with Modulus ". Toutefois, la version 13.0 a amené un algorithme nouveau et meilleur, le Mersenne Twister. Cet algorithme est assez robuste pour permettre à SPSS d'être utilisé pour des simulations Monte Carlo plus sérieuses.

Pour générer des valeurs à partir de SPSS, nous avons d'abord créé un support sur Excel, pour ensuite l'importer sur SPSS et générer 100 échantillons de 10.000 variables aléatoires suivant la loi Bernouilli(0.7) grâce à l'outil de génération de variables aléatoires de SPSS.



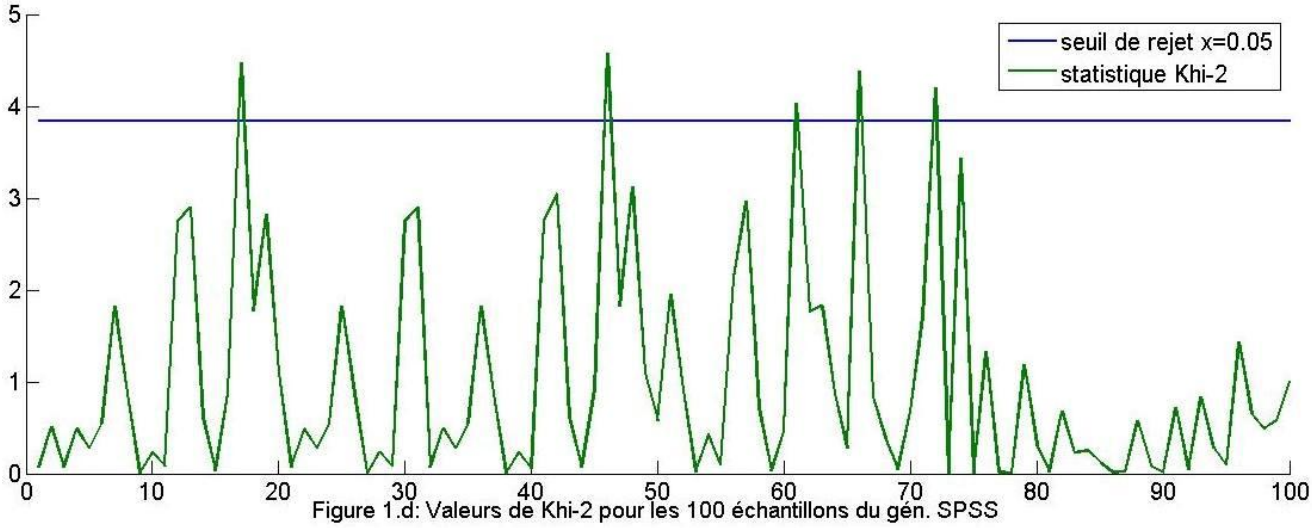


Figure 1.d: Valeurs de Khi-2 pour les 100 échantillons du gén. SPSS

D'après le résultat obtenu, le générateur SPSS présente 5 échantillons rejetés, et approximativement 10 échantillons tangents.

1.5.4. Test Khi-2 pour le générateur R:

A propos du générateur R:

R utilise un générateur congruentiel linéaire dont il génère la seed aléatoirement selon l'heure du système.

Comme pour les générateurs précédents, on génère 100 échantillons de 10000 variables aléatoires suivant la loi Bernouilli (0.7) grâce à la fonction ***** du langage R.

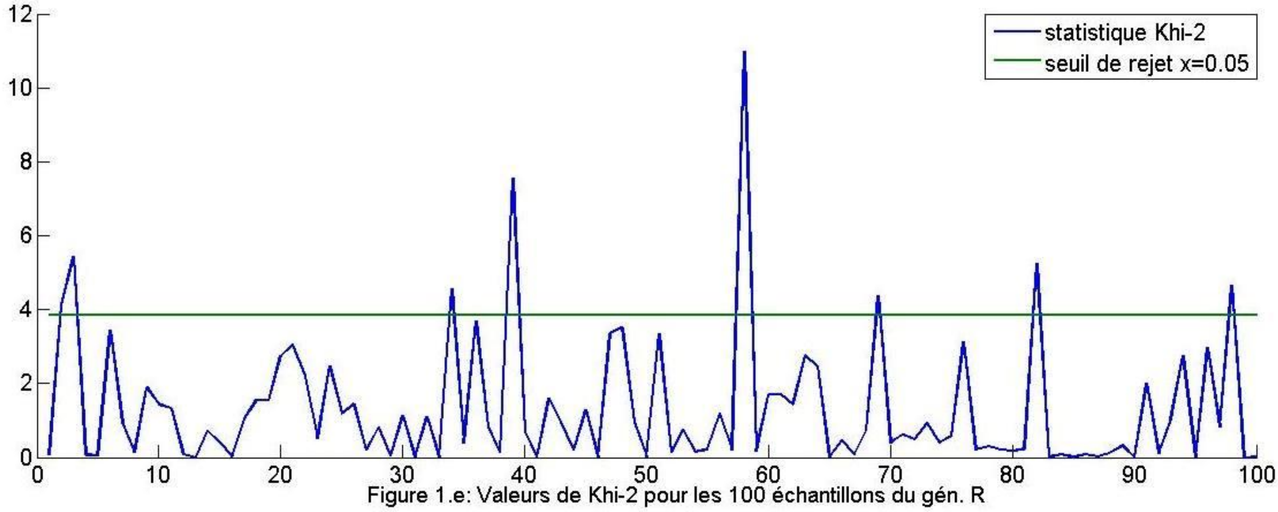


Figure 1.e: Valeurs de Khi-2 pour les 100 échantillons du gén. R

Le générateur R présente ainsi 8 échantillons rejetés et 11 échantillons tangents.

1.5.5. Test Khi-2 pour le générateur Hamlili :

Le générateur Hamlili a été mis au point par le Pr A. Hamlili. Son algorithme est le suivant :

```
tai=50000;a3=747391;a5=9648451;m=2^31-1;
x(1)=m/2;x(2)=m/3;x(3)=m/6;x(4)=m/4;x(5)=m/5;
i=6;
while i<=tai+6
    x(i)=mod(a5*x(i-5)+a3*x(i-3),m);
    i=i+1;
end;
x=x(6:tai+6)/m;
```

Ainsi, on répète la démarche similaire aux générateurs précédents et on applique le test de Khi-2 sur les 100 échantillons suivant la loi Bernouilli(0.7).

```
m=2^31-1;  x(1)=m/2;x(2)=m/3;x(3)=m/6;x(4)=m/4;x(5)=m/5;
i=6;
M=zeros(10000,100);
for j=1:100
    while i<=tai+6
        x(i)=mod(a5*x(i-5)+a3*x(i-3),m);
        i=i+1;
    end;
    M(1:10000,j)=x(7:tai+6)/m;
    for i=1:6
        x(7-i)=x(tai+7-i);
    end
 end
%Méthode des transfomations inverses
Val_ham_lin=zeros(10000,100);
for j=1:100
    nb=[0,0];
    for i=1:10000
        if M(i,j)<0.3
            Val_ber(i)=0;
            nb(1)=nb(1)+1;
        end
        if M(i,j)>=0.3
            Val_ber(i)=1;
            nb(2)=nb(2)+1;
        end
    end
    Var_ham_lin(1:10000,j)=Val_ber;
    freq_zero(j)=nb(1);
    freq_un(j)=nb(2);
end
for j=1:100 %calcul des statistiques
   s(j)=((freq_zero(j)-0.3*10000)^2)/(10000*0.3)+((freq_un(j)-0.7*10000)^2)/(10000*0.7);
end
```

Figure 1.f: Script utilisé pour implémenter le générateur Hamlili

Ainsi, on obtient la courbe suivante

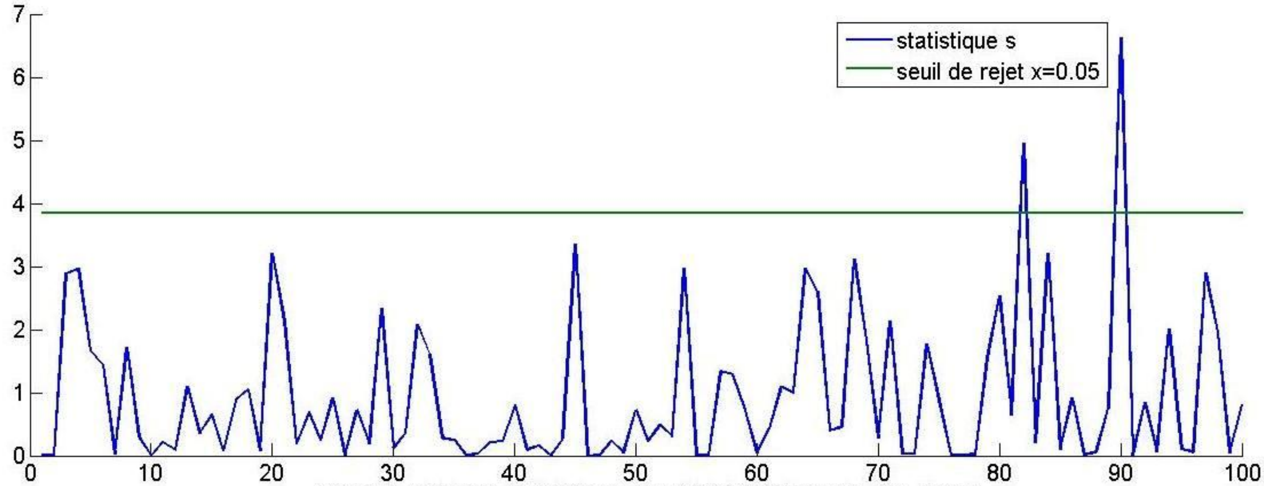


Figure 1.g: Valeurs de Khi-2 pour les 100 échantillons du gén. Hamlili

Le générateur Hamlili présente d'après les résultats affichés 6 échantillons rejetés et 4 échantillons tangents.

1.5.6. Test Khi-2 pour le générateur congruentiel linéaire :

Les nombres pseudo aléatoires forment une suite dont chaque terme dépend du précédent, selon la formule :

$$X_{n+1} = (aX_n + b)mod.m$$

où *a* est le *multiplicateur*, *b* l'*incrément* et *m* le *module*.

Le terme initial est appelé la graine (*seed* en anglais). C'est elle qui va permettre de générer une suite apparemment aléatoire. Pour chaque graine, on aura une nouvelle suite. Cependant, il est possible que certaines graines permettent d'obtenir une suite plus aléatoire que d'autres.

Du fait de l'opération *mod*, les termes de cette suite sont compris entre 0 et m. De plus, comme chaque terme dépend entièrement du précédent, si un nombre apparaît une deuxième fois, toute la suite se reproduit à partir de ce nombre. Or le nombre de valeurs que le nombre peut prendre étant fini (égal à m), la suite est amenée à se répéter au bout d'un certain temps. On dit qu'elle est ultimement périodique.

```
nb_ech=100; tai=50000; a=1103515245; b=12345; m=2^32; X=4294967291;
frequence=zeros(2,nb_ech);
gen_lin_ber=zeros(tai,nb_ech);
tab_des_u=zeros(tai,nb_ech);
for j=1:nb_ech
    nb=[0,0];
    for i=1:tai
        u=(mod(a*X+b,m));
        tab_des_u(i,j)=u; %génèration d'un nombre aléatoire par le gen linéaire
        %Méthode des transformations inverse
        if u/m<0.3
            Val_ber(i)=0;
            nb(1)=nb(1)+1;
        end
        if u/m>=0.3
            Val_ber(i)=1;
            nb(2)=nb(2)+1;
        end
        X=u;
    end
    fr(1:2,j)=nb;
    gen_lin_ber(1:tai,j)=Val_ber;
end
for j=1:nb_ech %Calcul de la statistique s pour les 100 échantillons
    s(j)=(fr(1,j)-0.3*tai)^2/(0.3*tai)+(fr(2,j)-0.7*tai)^2/(0.7*tai);
end
```

Figure 1.h: Script utilisé pour implémenter le générateur Linéaire

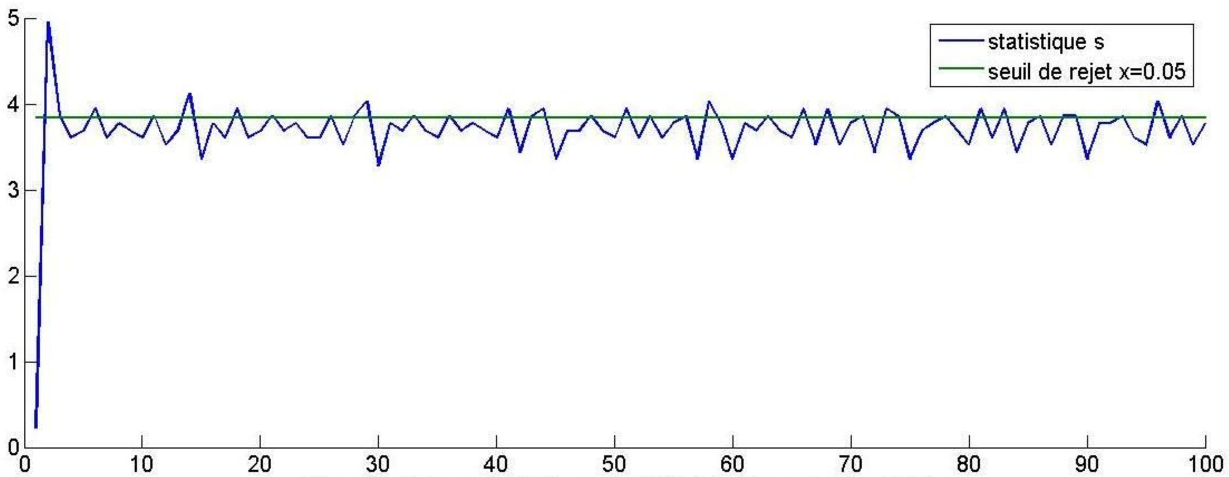


Figure 1.i: Valeurs de Khi-2 pour les 100 échantillons du gén. Linéaire

Le générateur linéaire présente 23 échantillons rejetés et 86 échantillons tangents (d'où son manque de qualité comparé aux autres générateurs).

1.5.7. Test Khi-2 pour le générateur multiple :

À propos du générateur multiple :

Le générateur multiple est similaire au générateur linéaire, sauf qu'il n'y a pas d'incrément. D'où sa formule :

$$X_{n+1} = (aX_n) mod.m$$

On effectue la même démarche qu'avec le générateur congruentiel linéaire.

```
nb_ech=100; tai=50000; X=4294967291; a=7^5; m=2^31-1;
frequence=zeros(2,nb_ech);
gen_lin_ber=zeros(tai,nb_ech);
tab_des_u=zeros(tai,nb_ech);
for j=1:nb_ech
    nb=[0,0];
    %génèration d'un nombre aléatoire par le gen multiple
    for i=1:tai
        u=(mod(a*X,m));
        tab_des_u(i,j)=u;
        %Méthode des transformations inverse
        if u/m<0.3
            Val_ber(i)=0;
            nb(1)=nb(1)+1;
        end
        if u/m>=0.3
            Val_ber(i)=1;
            nb(2)=nb(2)+1;
        end
        X=u;
    end
    fr(1:2,j)=nb;
    gen_lin_ber(1:tai,j)=Val_ber;
end
%Calcul de la statistique s pour les 100-échantillons
for j=1:nb_ech
    s(j)=(fr(1,j)-0.3*tai)^2/(0.3*tai)+(fr(2,j)-0.7*tai)^2/(0.7*tai);
end
```

Figure 1.j: Script utilisé pour implémenter le gén. Multiple

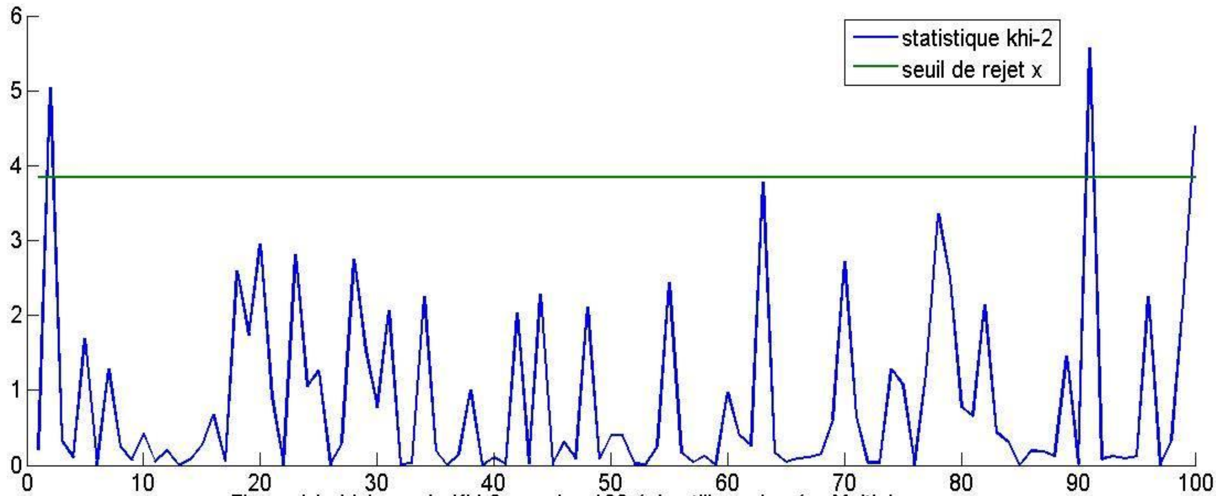


Figure 1.b: Valeurs de Khi-2 pour les 100 échantillons du gén. Multiple

Le générateur multiple présente dans ce résultat 2 échantillons rejetés et 4 échantillons tangents.

1.4. Effet de la graine (Seed) sur la qualité du générateur :

Pour évaluer l'effet du seed sur la qualité du générateur linéaire, on a essayé d'appliquer le test de Khi-2 sur plusieurs échantillons générérs par des gen. congruentiel linéaires de différents seed (des seeds à 10 chiffres, 8 chiffres, 12 chiffres).

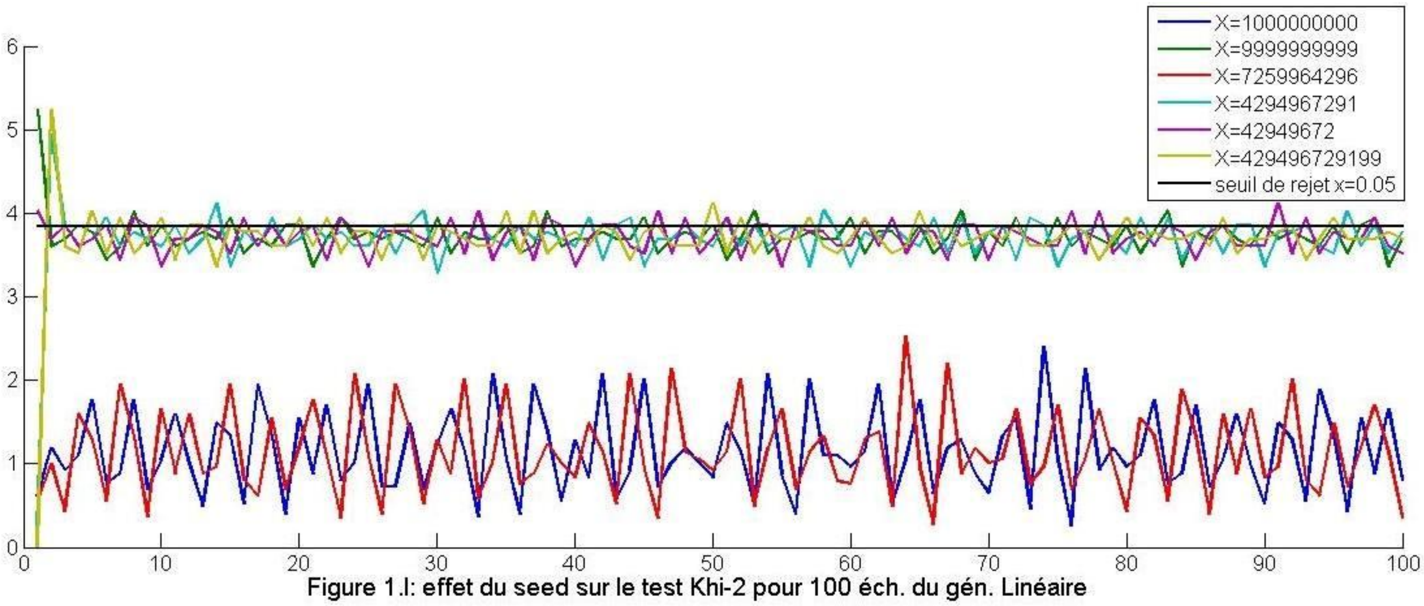


Figure 1.l: effet du seed sur le test Khi-2 pour 100 éch. du gén. Linéaire

Le résultat obtenu ne nous permet pas de trancher sur le domaine de valeurs des seed qui sont les plus performants. Toutefois, les valeurs « 10000000000» et « 7259964296 » ont des résultats plus valides que les autres valeurs prises dans le même test .

Chapitre 2:

Lois Continues, test KS, et corrélation avec test Khi-2

2. Lois continues, Test KS et corrélation avec test Khi-2:

2.1 : Définition:

Le test Kolmogorov-Smirnov est une opération sur un échantillon suivant une variable aléatoire permettant de décider s'il en est une bonne approximation. Pour ce faire, on a besoin d'ordonner les valeurs par ordre croissant, et de comparer chaque valeur à son emplacement dans l'échantillon (i/n et i-1/n). Le plus grand écart trouvé dans cette comparaison correspond à la valeur K, qui nous permettra de déterminer la qualité de l'approximation de la loi, et ce en la comparant à 1.36 / la racine de la taille de l'échantillon. Si cette valeur est supérieure à K, alors l'hypothèse S(U)=F(U) n'est pas rejetée.

$$H_0 \; : \; S(u) = F(u)$$
$$H_1 \; : \; S(u) \neq F(u)$$

Le test consiste à calculer $K = \max(K^+, K^-)$

$$K^+ = \max_{1 \le i \le n} \{ i/n - F(u_i) \} \qquad K^- = \max_{1 \le i \le n} \{ F(u_i) - (i-1)/n \}$$

2.2: Application à quelques générateurs:

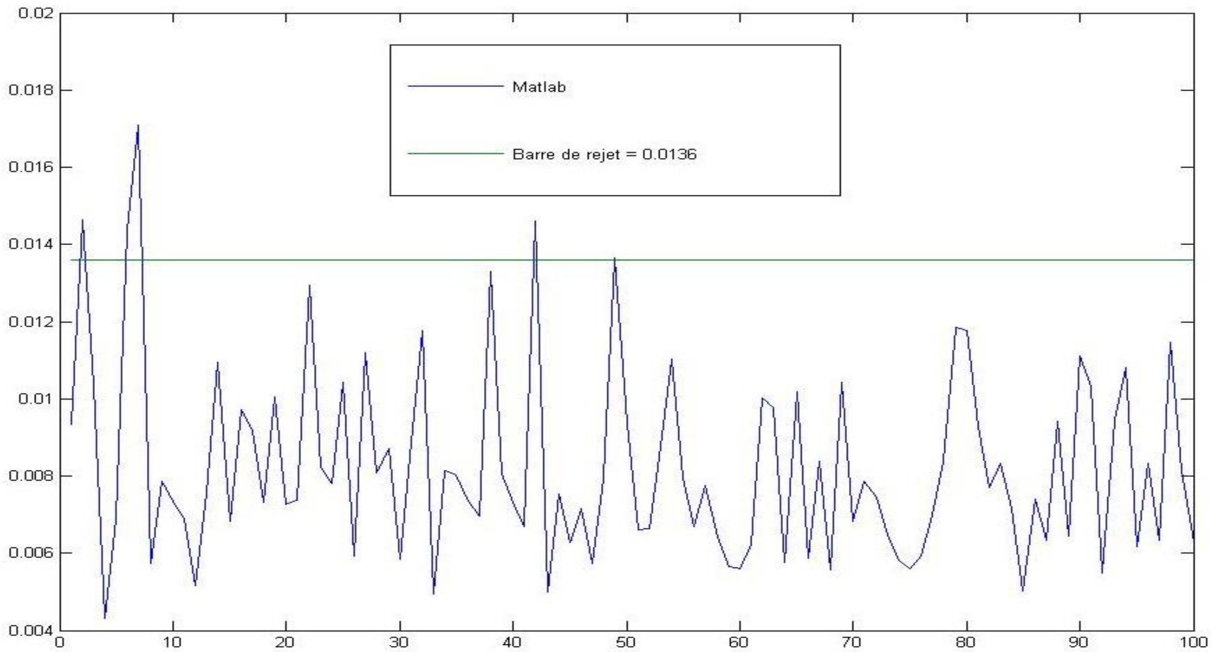


Figure 2.a: Test KS sur 100 échantillons du générateur Matlab

Notes: cette courbe représente les différentes valeurs de K pour 100 échantillons. Les 2 barres représentent $k_{0.05}$ et $k_{0.01}$ (pour nos calculs, on ne prendra en compte que $k_{0.05}$, qui est égale à $\frac{1.36}{\sqrt{taille(=10000)}}$.

Comme on peut le voir, il y a 5 échantillons qui passent la barre de $k_{0.05}$ qui sont non-acceptables. Les 95 autres échantillons sont de bonnes approximations de la loi uniforme.

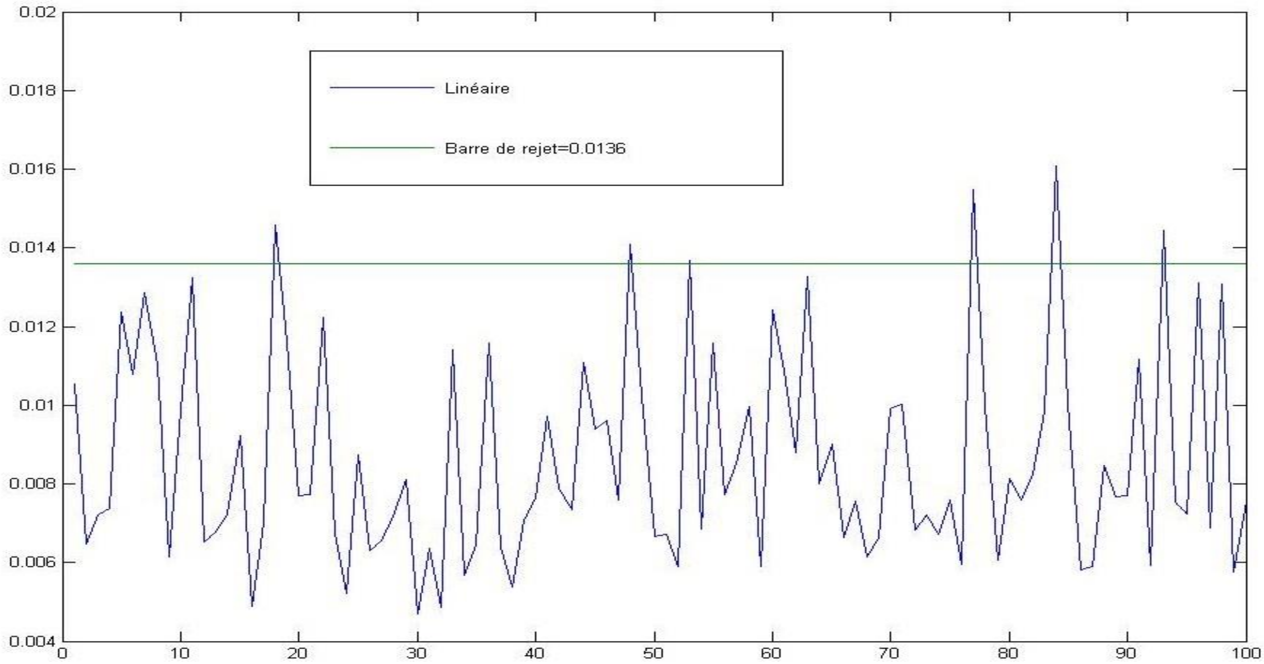


Figure 2.b: Test KS sur 100 échantillons du générateur Linéaire

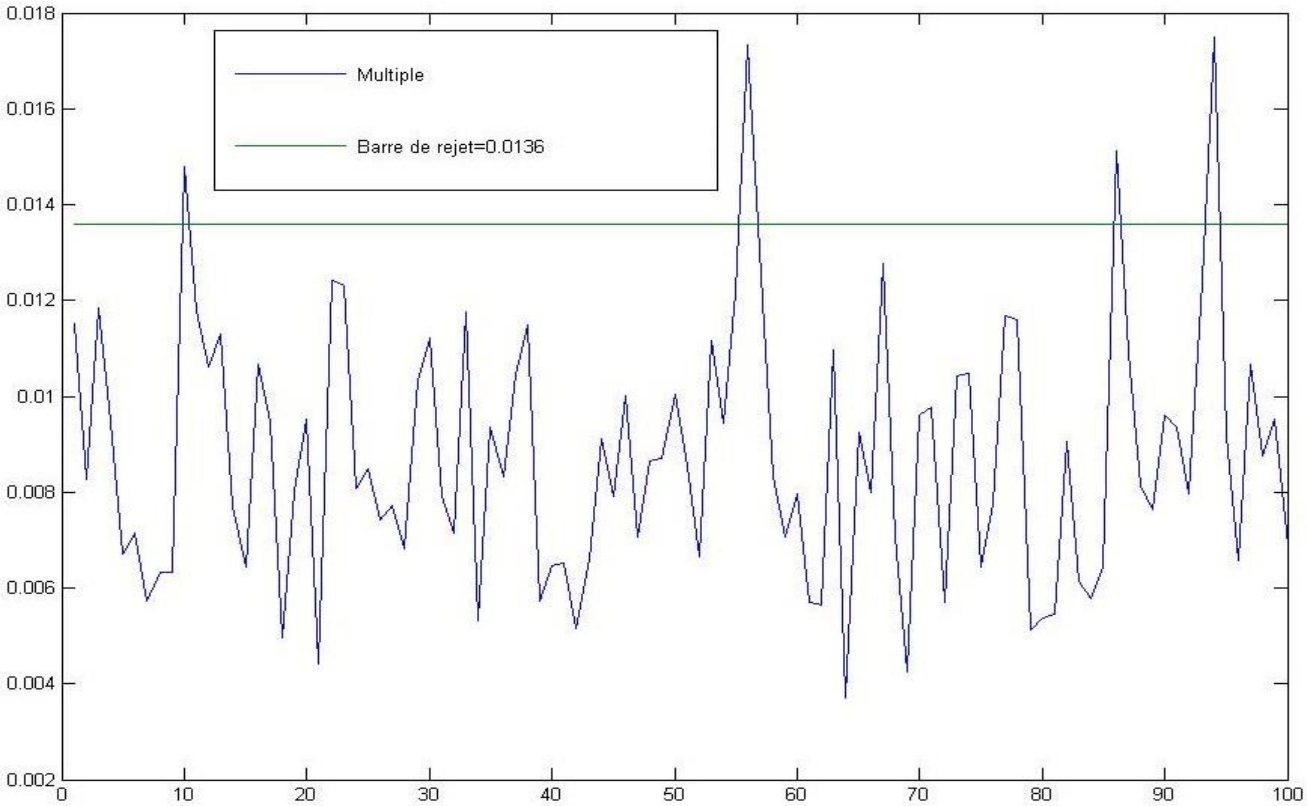


Figure 2.c: Test KS sur 100 échantillons du générateur Multiple

2.3: Parallèle entre Khi-2 et KS:

On souhaite maintenant établir un parallèle entre le test Khi-2 et le test KS. En effet, nous avons remarqué qu'il existait des pics dans les valeurs de khi-2 chez certains générateurs. Nous chercherons sont à trouver une corrélation entre des pics sur la courbe de khi-2 et sur la courbe de KS.

2.3.1: Méthodologie:

Pour réaliser cela, nous allons générer des échantillons de la loi uniforme sur lesquels nous appliquerons le test KS, et nous allons nous servir de la transformée inverse de la loi uniforme pour générer une loi binomiale sur laquelle nous appliquerons le test Khi-2.

2.3.2: Résultats:

Par souci de clarté, nous avons amplifié la valeur de K 500 fois pour qu'elle soit à la même échelle que Khi2 sur le graphe. Voici les résultats:

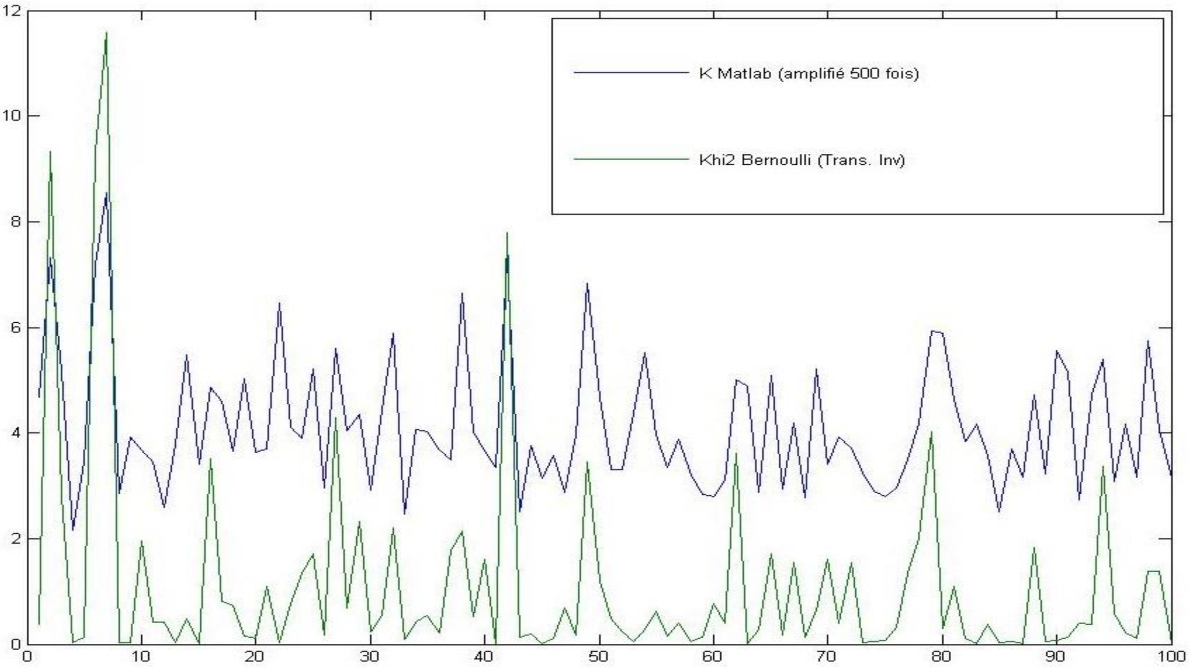


Figure 2.d: Comparaison de K et Khi sur 100 échantillons du générateur Matlab

On voit clairement une concordance entre les pics des 2 courbes. On peut donc conclure que les pics constatés dans les courbes étudiées de Khi2 sont causées par une grande valeur de K pour la loi uniforme associée.

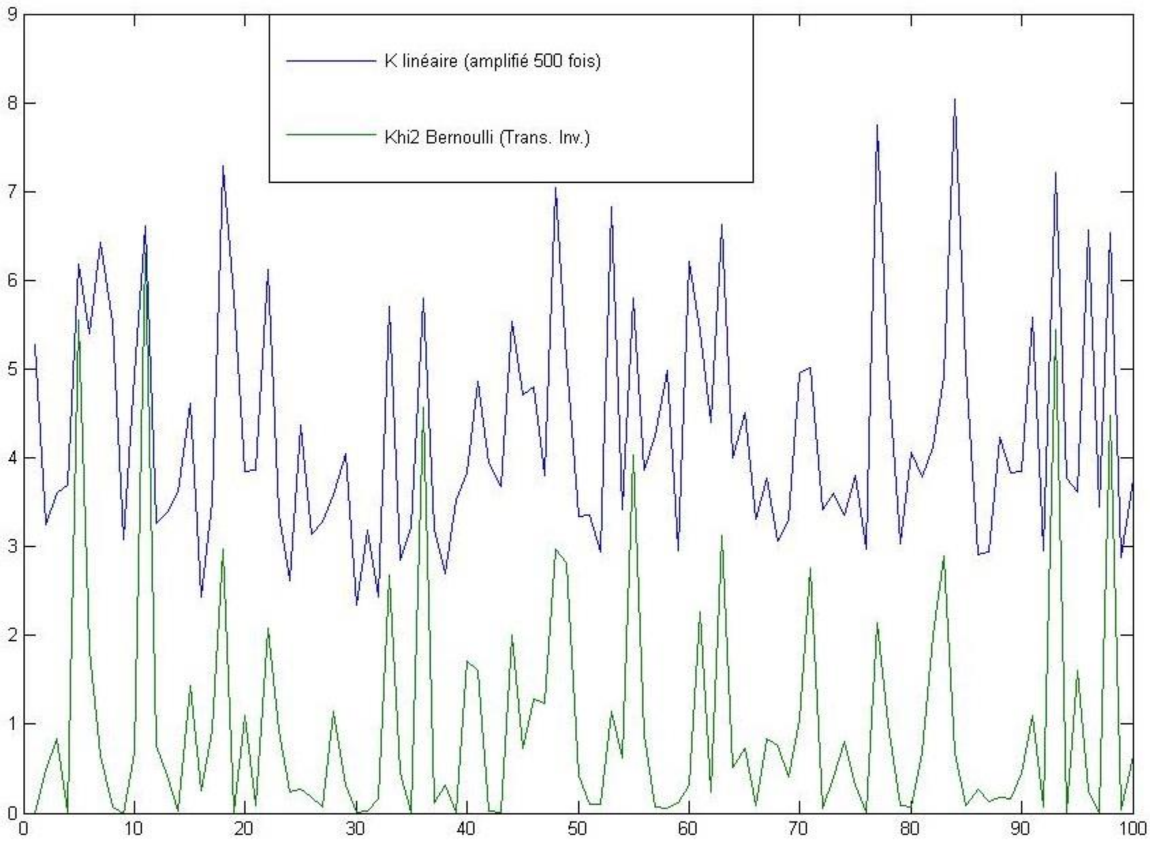


Figure 2.e: Comparaison de K et Khi sur 100 échantillons du générateur Linéaire

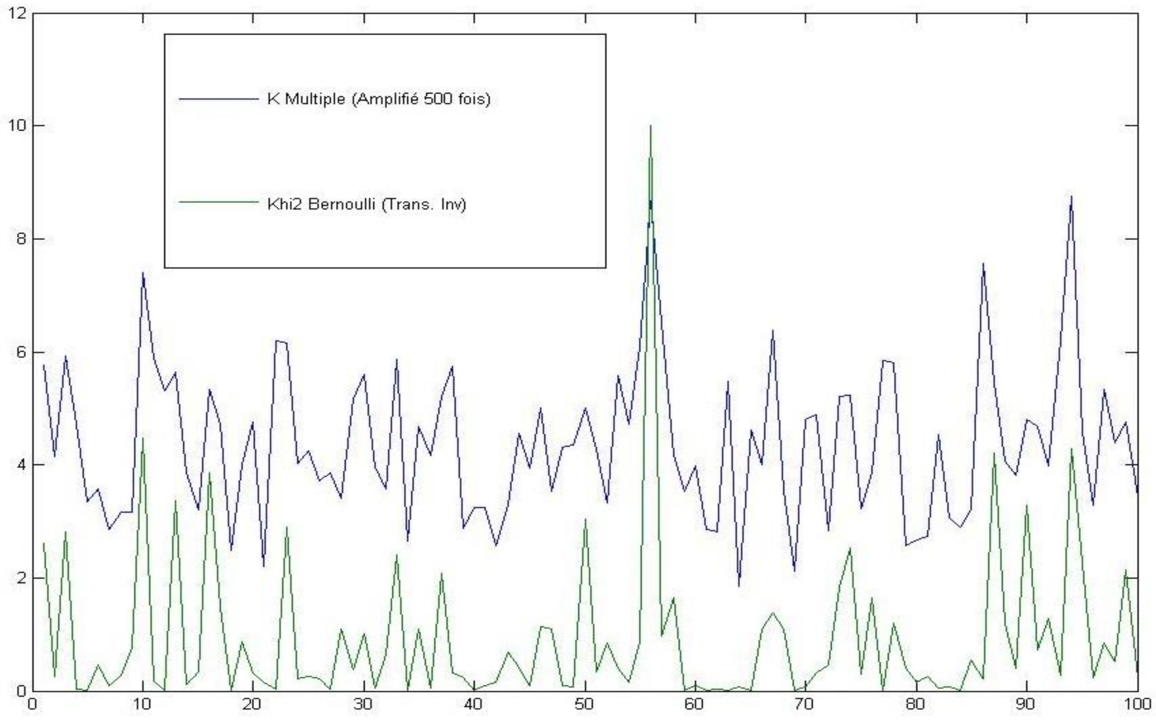


Figure 2.f: Comparaison de K et Khi sur 100 échantillons du générateur Multiple

Chapitre 3:

Etude du test KS à la lumière de la discrépance

3. Etude du test KS à la lumière de la discrépance:

3.1. But de l'étude:

Nous allons nous intéresser aux valeurs de K pour chaque échantillon, et nous demander ce qui cause la hausse de cette valeur. Penchons nous sur la discrépance.

3.2: Définition de la discrépance:

La discrépance est la propriété d'une suite à diverger d'une suite de même taille équi-répartie. Par exemple, pour un intervalle [0,1], un ensemble équiréparti d'une valeur se trouverait au point 0.5, de 2 valeurs se trouverait aux points 0.33 et 0.66, etc. Plus une suite de nombres est équirépartie, plus sa discrépance n'est faible. Nous cherchons donc à démontrer que l'augmentation de la discrépance entraine une augmentation de la valeur de K.

3.3: Etudes parallèles de la discrépance et de K:

Voici la démarche que l'on va suivre pour essayer de démontrer cette corrélation: grâce à un générateur linéaire, on va générer n échantillons de k valeurs (on va prendre le cas (n,k)=(500,500)). S'il y a m échantillons au dessus de la limite, on prendre les m meilleurs échantillons, les m pires, les m pires parmi les tolérables, et l'on va étudier la répartition de leurs valeurs. Pour cela, nous allons diviser l'intervalle [0,1] en 10 sous-intervalles, et on va dénombrer l'effectif de chaque sous-intervalle pour chaque échantillon, le but étant de prouver que plus un échantillon est bon, plus l'écart entre l'effectif de l'échantillon le plus peuplé et le moins peuplé est faible.

3.4: Résultats:

Les résultats qu'on a trouvés sont mitigés. Malgré une certaine corrélation au niveau de l'écart entre le min et le max des écarts (en moyenne 14 pour les meilleurs, 24 pour les tangents et 28 pour les rejetés), on retrouve tout de même certains échantillons rejetés qui partagent des propriétés avec des échantillons acceptés, ce qui ne nous permet pas de tirer une conclusion.

minmaxbest =

43	54
45	57
43	63
45	54
42	61
44	55
42	57
43	55
43	57
39	59
42	59
41	56
43	55
43	57
47	55
42	59
44	57
42	57
41	59
39	62
42	58
45	55
43	57
43	56
41	58
43	59
42	55
44	55

intervallebest10 =

51	54	48	53	43	53	51	48	54	45
49	46	53	47	57	50	53	45	50	50
49	50	44	63	43	57	47	49	53	45
50	49	54	51	52	49	45	52	49	49
44	60	44	49	53	48	52	47	61	42
50	52	53	46	55	48	48	44	54	50
53	48	42	53	53	48	46	53	47	57
44	51	55	53	50	48	50	54	43	52
50	49	47	54	57	46	55	47	52	43
53	44	59	39	58	43	49	51	54	50
59	42	50	43	53	52	46	51	55	49
50	55	45	49	54	41	56	48	50	52
54	55	43	52	54	48	48	54	46	46
53	48	47	50	51	43	57	48	53	50
51	47	51	55	50	47	52	52	48	47
43	53	51	51	49	59	42	49	54	49
51	50	52	54	44	45	52	47	57	48
46	48	57	47	55	42	51	57	42	55
54	44	48	51	59	50	41	51	52	50
52	39	62	41	50	51	60	44	51	50
44	58	55	44	48	57	51	42	49	52
53	50	45	51	48	50	48	55	51	49
50	44	50	52	53	47	50	43	57	54
56	47	49	43	53	53	54	51	51	43
49	46	52	58	51	41	50	45	57	51
45	47	54	45	59	56	47	52	52	43
51	52	44	52	52	47	51	55	54	42
55	45	54	45	53	51	44	52	50	51

Figure 3.a: valeurs des sous-effectifs des 28 meilleurs échantillons

minmaxtoler =

36	61
41	65
37	62
40	60
40	68
36	60
36	62
38	72
43	61
38	64
40	68
42	71
30	60
40	67
41	63
41	64
38	62
36	59
42	57
34	63
38	60
42	60
39	63
40	63
40	62
36	61
39	61
39	56

intervalletoler10 =

36	51	42	49	55	51	55	44	61	56
53	50	65	49	42	53	47	50	50	41
55	37	42	48	50	46	57	49	54	62
53	56	60	49	48	40	45	55	54	40
51	47	48	47	68	53	60	40	40	46
60	57	57	47	45	48	49	36	55	46
44	46	42	62	44	36	58	58	51	59
38	48	39	72	50	63	45	44	52	49
43	46	54	61	57	60	43	46	45	45
62	41	64	56	45	47	55	48	38	44
47	40	49	43	47	58	52	45	68	51
47	46	43	49	46	42	57	49	71	50
59	52	60	52	47	54	30	52	48	46
64	40	47	49	56	67	43	42	46	46
46	42	41	47	63	43	53	58	53	54
64	52	52	52	44	53	41	45	44	53
38	58	49	42	47	46	44	57	62	57
36	46	47	59	53	51	56	52	56	44
56	56	54	57	48	54	44	46	43	42
50	40	50	63	58	60	54	34	47	44
48	51	54	54	56	60	47	49	38	43
51	57	49	60	47	44	45	42	50	55
42	49	47	54	44	39	52	56	63	54
42	56	47	41	52	40	48	49	62	63
49	40	46	40	50	59	62	46	57	51
60	36	45	44	47	42	61	54	61	50
42	39	44	56	55	53	61	53	52	45
55	56	49	56	54	50	53	39	42	46

Figure 3.b: valeurs des sous-effectifs des 28 échantillons tolérables

intervalleover10 =

50	51	44	61	52	57	59	52	41	33
45	35	45	47	60	44	60	55	47	62
52	40	34	50	63	51	58	53	43	56
50	49	49	59	51	61	52	53	36	40
54	71	51	42	40	49	52	46	48	47
43	41	36	75	51	64	51	51	44	44
33	51	54	50	39	48	46	59	61	59
43	45	42	39	53	61	58	49	50	60
59	40	62	57	58	52	36	38	54	44
50	44	50	63	61	46	66	35	36	49
54	61	57	58	47	42	45	46	42	48
41	47	45	58	34	47	55	58	57	58
29	52	39	56	57	48	50	46	66	57
66	59	51	48	41	49	46	44	53	43
60	49	51	57	52	61	35	44	37	54
38	46	54	32	64	52	51	61	61	41
39	54	55	33	40	58	58	51	58	54
50	32	42	50	61	49	45	49	63	59
55	33	49	46	43	43	54	60	56	61
56	48	51	64	53	59	39	45	40	45
51	47	36	47	45	47	56	56	64	51
58	54	61	51	50	58	40	36	52	40
47	42	37	61	38	49	42	61	69	54
54	59	55	58	57	44	36	48	49	40
45	58	36	51	38	50	35	69	50	68
61	56	55	65	42	57	41	45	38	40
52	76	57	48	56	45	44	49	29	44
40	34	40	55	53	38	54	63	58	65

minmaxover =

33	61
35	62
34	63
36	61
40	71
36	75
33	61
39	61
36	62
35	66
42	61
34	58
29	66
41	66
35	61
32	64
33	58
32	63
33	61
39	64
36	64
36	61
37	69
36	59
35	69
38	65
29	76
34	65

Figure 3.c: valeurs des sous-effectifs des 13 échantillons non-acceptés

4. Conclusion Générale:

Ce projet visait à appliquer des algorithmes de test sur des échantillons de générateurs afin de quantifier leur qualité. On a réussi à établir une corrélation entre les valeurs de Khi-2 et de K, ainsi qu'à montrer qu'il existe un lien entre la discrépance d'un échantillon et le résultat du test KS.

Ce faisant, nous avons acquis la capacité à comprendre ce qui constitue un bon générateur, chose qui nous paraissait très technique et difficilement abordable, une bonne maitrise des logiciels utilisés pour mener à bien notre tâche à force de répétition et d'application, et une capacité à gérer les deadlines et à concilier l'avancement d'un projet avec les contraintes d'une période d'examens.

5. Bibliographie:

5.1. Cours:

- Pr. Ibrahim Amrani, "*Génération de nombres au hasard*",
- Pr. Ibrahim Amrani, "*Génération de lois à densité*",
- support.office.com/en-us/article/rand-function…
- http://www.groupes.polymtl.ca/geo/marcotte/mth2302c/chapitre7

5.2. Logiciels:

- Matlab [Logiciel], version R2013a, développé par MathWorks, sorti en 1984, https://fr.mathworks.com
- Excel, [Logiciel], version 2013, développé par Microsoft, sorti en 1985, products.office.com
- SPSS Statistics [Logiciel], version 17.0, développé par SPSS, sorti en 1968, www.ibm.com/software/analytics/spss/products/statistics/index.html
- R Studio [Logiciel], version 3.4.4, développé par R-Tools Technology, sorti en 2011, http://www.r-studio.com/