Université des Sciences et Technologies de Lille
U.F.R. de Mathématiques Pures et Appliquées
Bât. M2, F-59655 Villeneuve d'Ascq Cedex

Agrégation externe **Année 2003-2004**

Corrigé de l'exercice sur le test des signes

Ex 1. *Test des signes (observations couplées)* [1, p. 112]

Pour essayer un nouvel engrais B devant améliorer le rendement en blé, nous disposons de n champs d'une station expérimentale (n petit de l'ordre de 10) ; chacun de ces champs étant divisé en deux parcelles. Affectons avec la probabilité 1/2 l'engrais B à une parcelle d'un champ et l'engrais usuel A à l'autre parcelle. Pour le champ numéro i, on note (X_i, Y_i) les rendements respectifs des parcelles ayant reçu A et B. On se propose de tester

$$(\mathcal{H}_0) \quad \text{« } B \text{ et } A \text{ donnent le même rendement »}$$
contre
$$(\mathcal{H}_1) \quad \text{« } B \text{ améliore le rendement ».}$$

Considérons alors sous $(\mathcal{H}_0)$, la probabilité d'avoir $\{X_i < Y_i\}$. Comme les engrais ont été affectés avec la probabilité 1/2 à chacune des parcelles, $\mathbf{P}(X_i > Y_i) = \mathbf{P}(X_i < Y_i)$. Supposons que $\mathbf{P}(X_i = Y_i)$ est nulle : alors les v.a. $U_i = \mathbf{1}_{\{X_i<Y_i\}}$ sont des v.a. de Bernoulli indépendantes de paramètre 1/2 et $S = U_1 + \cdots + U_n$ suit la loi $\mathrm{Bin}(n, 1/2)$. Sous l'hypothèse $(\mathcal{H}_1)$, on observera plus souvent $\{X_i < Y_i\}$ et S sera plus grand. D'où la région de rejet du *test du signe* : on définit s_α comme le plus petit entier s tel que

$$\sum_{k=s+1}^{n} \frac{C_n^k}{2^n} \leq \alpha,$$

et on prend comme région de rejet $D_\alpha = \{S > s_\alpha\}$.

1) Déterminer les valeurs de s_α en fonction de α pour $n = 10$. On inversera directement la f.d.r. de la binomiale, sans faire confiance à `cdfbin`.

2) Simuler les rendements sous $(\mathcal{H}_0)$ et sous $(\mathcal{H}_1)$ et mettre en œuvre le test.

3) Construire un test *randomisé* permettant d'avoir exactement le niveau α.

Corrigé

1) Voici deux fonctions Scilab (empruntées à D. Flipo) répondant à la question

```
function [s]=testsgn(n,a)
// La fonction testsgn retourne le plus petit entier s tel que
// \sum_{k=s+1}^n C_n^k <= 2^n a, (avec une boucle while).
// Pour cela on calcule les coefficients binomiaux C_n^k,
// la ligne 'n' du triangle de Pascal est donnée par :
C=[1];
for i=1:n, C=[C,0]+[0,C];end  // Attention : C_n^k = C(k+1)
// Méthode élémentaire (avec boucle while)
b= a * 2^n;
S=1;s=n;
while S <= b, S=S+C(s) ; s=s-1; end;
endfunction

function [s]=testsgn1(n,a)
// La fonction testsgn1 retourne également le plus petit entier s tel que
// \sum_{k=s+1}^n C_n^k <= 2^n a, à partir de cumsum() sans boucle.
// Pour cela on calcule les coefficients binomiaux C_n^k,
// la ligne 'n' du triangle de Pascal est donnée par :
C=[1];
for i=1:n, C=[C,0]+[0,C];end  // Attention : C_n^k = C(k+1)
// Méthode utilisant cumsum() : on profite du fait que
// les coefficients binomiaux sont symétriques (C_n^k = C_n^{n-k}).
S=cumsum(C);
s=n-max(find(S <= a * 2^n));
endfunction
```

On obtient ainsi

```
 -->testsgn(10,0.05)
 ans  =

    8.
```

On peut aussi traiter le problème de façon un peu plus générale en remarquant que le s_α cherché est l'entier minimal s tel que $\mathbf{P}(S > s) \leq \alpha$ et que cette condition équivaut à $\mathbf{P}(S \leq s) \geq 1 - \alpha$. Par conséquent, la f.d.r. étant discontinue seulement aux points entiers $k = 0, \dots, n$ et constante sur les intervalles $[k, k+1[$,

$$s_\alpha = \min\{s \in \mathbb{N};\ \mathbf{P}(S \leq s) \geq 1 - \alpha\} = \inf\{t \in \mathbb{R};\ \mathbf{P}(S \leq t) \geq 1 - \alpha\},$$

ce qui montre que s_α est exactement l'image de $1 - \alpha$ par l'inverse généralisé de la f.d.r. de S. On peut alors exploiter la fonction `invfdrbin` de `Binomiales.sci`, fournie avec ce document.

```
-->;getf("/home/suquet/Enseignement/Scilab/Binomiales.sci");
```

```
-->alpha=0.05;n=10;salpha=invfdrbin(n,0.5,1-alpha)
 salpha  =

    8.
```

2) Pour mettre en oeuvre le test, plaçons nous d'abord sous l'hypothèse nulle. Il est raisonnable de supposer que les rendements sont des variables aléatoires gaussiennes, mais comme les caractéristiques de chaque champ peuvent influer sur les rendements, nous allons prendre des paramètres différents pour chaque champ. Disons pour fixer les idées que l'on choisit ces paramètres au hasard en prenant pour le i-ème champ une moyenne $m_i = 10 + 3V_i$ où V_i suit une loi uniforme sur $[0, 1]$ et un écart-type $\sigma_i = 2 + W_i$ où W_i suit une loi uniforme sur $[0, 1]$.

Voici un script relativement simple (l'affichage des rendements sous forme de tableau peut être sauté en première approche...).

```
// Test des signes en agronomie
// simulation des rendements des parcelles sous deux engrais
// et réalisation du test déterministe
n=10;
alpha=0.05;
// On suppose les rendements gaussiens de paramètres m et sigma
// choisis comme suit
moy=10+3.*rand(1:n,'uniform');
sig=2+rand(1:n,'uniform');
Z=rand(2,n,'normal');
X=moy+sig.*Z(1,:);
Y=moy+sig.*Z(2,:);
// affichage des rendements
printf('%6s%13s%13s','Champ','Rendement A','Rendement B')
printf('%6d%13.2f%13.2f\n',(1:n)',X',Y')
//
// Calculs pour le test:
U=bool2s(X<Y);
S=sum(U);
salpha=invfdrbin(n,0.5,1-alpha);
// Cette instruction suppose que l'on ait déjà chargé Binomiales.sci
// à adapter donc!
if S<=salpha then
   disp("Acceptation de l''hypothèse de non amélioration du rendement")
else
   disp("Rejet de l''hypothèse de non amélioration du rendement")
end
```

À l'exécution, cela donne :

```
-->;exec("/home/suquet/Enseignement/Scilab/Tests/rendementH0.sce");
 Champ  Rendement A  Rendement B
     1        11.59        13.48
     2        13.36         9.04
     3         7.88         3.50
     4        18.52        11.88
     5         7.92        10.68
     6         9.21         8.18
     7         9.88        15.55
     8        11.14         9.59
     9        13.01        16.32
    10         9.16        10.90

 Acceptation de l'hypothèse de non amélioration du rendement
```

Intermède ludique. J'avoue ne pas avoir résisté à la tentation d'enjoliver un peu l'affichage des rendements comparés en rajoutant une colonne de signes (ce qui explique le nom du test !) : `'+'` si $X_i < Y_i$ et `'-'` sinon. C'est sans doute un luxe dangereux à déconseiller pour le jour de l'oral. Considérez plutôt cet amendement comme un petit exercice sur les matrices de chaînes de caractères et l'indexation des matrices par des booléens. Pour réaliser cet amendement, il suffit de remplacer les deux lignes d'affichage des rendements par ce qui suit :

```
// affichage des rendements
// avec colonne de signes '+' si X<Y, '-' sinon
//
// initialisation d'une colonne de '+'
plus='+';
col=plus(ones(n,1));
// modification par indexation booléenne
col(X'>=Y')='-';
//
printf('%6s%13s%13s%6s','Champ','Rendement A','Rendement B','Gain')
printf('%6d%13.2f%13.2f%6s\n',(1:n)',X',Y',col)
//
```

Voici un exemple d'exécution (le nouveau script a changé de nom) :

```
-->;exec("/home/suquet/Enseignement/Scilab/Tests/rendsignH0.sce");
 Champ  Rendement A  Rendement B  Gain
     1        10.98        11.75     +
     2         7.32         6.39     -
     3         9.09         7.53     -
     4        14.36        14.61     +
     5        10.63        15.36     +
     6         9.27         8.25     -
     7         6.54         9.34     +
     8        13.35         4.42     -
     9        11.36        12.95     +
    10        12.47        13.48     +

 Acceptation de l'hypothèse de non amélioration du rendement
```

Pour simuler les rendements sous $(\mathcal{H}_1)$, on peut se contenter d'augmenter la moyenne de Y d'un facteur fixe. Par exemple on peut remplacer la définition de `Y` dans le script par

```
Y=1.25.*moy + sig.*Z(2,:); // augmentation de 25% du rendement moyen
```

En faisant tourner le script sous $(\mathcal{H}_0)$, on constate que le rejet de l'hypothèse nulle est très rare. Cela n'est en fait pas surprenant car bien que l'on ait choisi $\alpha = 0.05$, la probabilité d'erreur de première espèce[1] (rejet de $(\mathcal{H}_0)$ alors qu'elle est vraie) vaut $\mathbf{P}(S > 8) \simeq 0.0107422$. Pour avoir un test de niveau 0.05, nous allons construire un test *randomisé*.

3) Rappelons qu'une statistique de test φ est une application mesurable $\Omega \to [0, 1]$, donc en pratique une fonction mesurable du vecteur échantillon observé. Le test associé à φ est défini par la règle de décision suivante :
- Si $\varphi(\omega) = 0$, on accepte $(\mathcal{H}_0)$.
- Si $\varphi(\omega) = 1$, on rejette $(\mathcal{H}_0)$.
- Si $\varphi(\omega) = p \in]0, 1[$, on rejette $(\mathcal{H}_0)$ avec probabilité p et on l'accepte avec probabilité $1 - p$.

On dit que le test est *déterministe* si φ ne peut prendre que les valeurs 0 et 1. Sinon, on parle de test *randomisé* ou *stochastique*. Dans le cas d'un test déterministe, on définit la région critique ou région de rejet $R := \varphi^{-1}(\{1\}) = \{\omega \in \Omega;\ \varphi(\omega) = 1\}$. Clairement dans ce cas, la fonction φ peut s'écrire $\varphi = \mathbf{1}_R$. Par contre dans le cas d'un test randomisé, φ n'est pas une fonction indicatrice (elle prend au moins une autre valeur que 0 et 1) et on ne peut parler de région de rejet.

1. Bien que $(\mathcal{H}_0)$ soit une hypothèse composite correspondant à une infinité de choix de lois possibles pour les rendements des parcelles, on peut calculer cette probabilité car la loi de S est toujours sous $(\mathcal{H}_0)$ la binomiale $\mathrm{Bin}(n, 0.5)$, pourvu que les champs soient indépendants et que pour chaque i, X_i ait même loi que Y_i avec $\mathbf{P}(X_i = Y_i) = 0$.

Pour construire un test des signes randomisé ayant exactement le niveau α, on part du test déterministe étudié ci-dessus et dont la fonction de test s'écrit :

$$\varphi(\omega) = \mathbf{1}_{D_\alpha}(\omega) = \begin{cases} 1 & \text{si } S(\omega) > s_\alpha, \\ 0 & \text{si } S(\omega) \leq s_\alpha, \end{cases}$$

avec s_α défini par

$$\mathbf{P}(S > s_\alpha) \leq \alpha < \mathbf{P}(S \geq s_\alpha).$$

Avec ce test, la probabilité de rejetter $(\mathcal{H}_0)$ alors qu'elle est vraie est $\mathbf{P}(S > s_\alpha)$ qui est seulement *majorée* par α. Pour rendre cette probabilité de rejet exactement égale à α, on introduit le test de fonction ψ définie par :

$$\psi(\omega) = \begin{cases} 1 & \text{si } S(\omega) > s_\alpha, \\ 0 & \text{si } S(\omega) < s_\alpha, \\ p & \text{si } S(\omega) = s_\alpha, \end{cases}$$

où $p \in]0,1[$ est ajusté de façon que la probabilité sous $(\mathcal{H}_0)$ de rejet de $(\mathcal{H}_0)$ par le test ψ soit exactement α. Cette contrainte s'écrit :

$$\mathbf{P}(S > s_\alpha) + p\mathbf{P}(S = s_\alpha) = \alpha,$$

d'où

$$p = \frac{\alpha - \mathbf{P}(S > s_\alpha)}{\mathbf{P}(S = s_\alpha)}.$$

La réalisation pratique du test de fonction ψ est très simple. On procède comme pour φ, mais lorsque $S = s_\alpha$, on génère une variable aléatoire V de loi uniforme sur $[0,1]$ et on décide de rejetter $(\mathcal{H}_0)$ si $V < p$ (donc avec probabilité p) et de l'accepter sinon. Voici un script possible.

```
// Test des signes randomisé en agronomie
// simulation des rendements des parcelles sous deux engrais A et A
// et réalisation du test de niveau exact alpha
// sous H0: pas d'amélioration rendement
n=10;
alpha=0.05;
// On suppose les rendements gaussiens de paramètres m et sigma
// choisis comme suit
moy=10+3.*rand(1:n,'uniform');
sig=2+rand(1:n,'uniform');
Z=rand(2,n,'normal');
X=moy+sig.*Z(1,:);
Y=moy+sig.*Z(2,:);
// affichage des rendements
// avec colonne de signes '+' si X<Y, '-' sinon
```

```
//
// initialisation d'une colonne de '+'
plus='+';
col=plus(ones(n,1));
// modification par indexation booléenne
col(X'>=Y')='-';
//
printf('\n%6s%13s%13s%6s','Champ','Rendement A','Rendement B','Gain')
printf('%6d%13.2f%13.2f%6s\n',(1:n)',X',Y',col)
printf('\n%s%.2f',' Test des signes randomisé, au niveau alpha=',alpha)
//
// Calculs pour le test:
U=bool2s(X<Y);
S=sum(U);
salpha=invfdrbin(n,0.5,1-alpha);
b=loibin(n,0.5);   // ainsi b(1+salpha)=P(S=salpha)
p=(alpha-1+fdrbin(n,0.5,salpha))./ b(1+salpha);
// Ces instructions supposent que l'on ait déjà chargé Binomiales.sci
// à adapter donc!
rejet="Rejet de l''hypothèse de non amélioration du rendement";
acceptation="Acceptation de l''hypothèse de non amélioration du rendement";
//
if S<salpha then
   disp(acceptation)
elseif S>salpha then
   disp(rejet)
else                                    // cas S=salpha
      if rand(1,1,'uniform')<p then     // rejet avec proba p
         disp(rejet)
      else                              // acceptation avec proba 1-p
         disp(acceptation)
      end
end
```

Bien sûr en exécutant plusieurs fois le script ci-dessus, il n'est pas aisé de voir si l'on a effectivement amélioré les choses. Pour cela, on écrit un nouveau script permettant d'exécuter les deux tests un grand nombre N de fois sur le même jeu de données et on observe les fréquences empiriques d'erreur de première espèce pour chacun des deux tests. C'est le but du script suivant que l'on vous laisse étudier.

```
// Tests des signes déterministe et randomisé en agronomie
// simulation des rendements des parcelles sous deux engrais A et A
// et réalisation du test de niveau exact alpha
// sous H0: pas d'amélioration rendement
```

```
// on cherche la fréquence empirique de rejet à tort de H0
n=10;
alpha=0.05;
N=5000; // nombre de tests simulés
// On suppose les rendements gaussiens de paramètres m et sigma
// choisis comme suit
moy=10+3.*rand(1:n,'uniform');
sig=2+rand(1:n,'uniform');
X=zeros(N,n);Y=zeros(N,n);
Z=rand(N,2*n,'normal');
for j=1:n, X(:,j)=moy(j)+sig(j).*Z(:,j); end
for j=1:n, Y(:,j)=moy(j)+sig(j).*Z(:,n+j); end
// Calculs pour le test:
U=bool2s(X<Y);
S=sum(U,'c');
salpha=invfdrbin(n,0.5,1-alpha);
b=loibin(n,0.5);   // ainsi b(1+salpha)=P(S=salpha)
p=(alpha-1+fdrbin(n,0.5,salpha))./ b(1+salpha);
// Ces instructions supposent que l'on ait déjà chargé Binomiales.sci
// Test des signes déterministe
testdet=(S>salpha);  // vecteur booleen (N,1), T= rejet, F=acceptation
ftd=sum(bool2s(testdet))./N;  // fréquence erreurs 1ère espèce
// Test des signes randomisé
V=rand(S,'uniform');
testrand=(S>salpha)|((S==salpha)&(V<p));  // attention == et pas =
ftr=sum(bool2s(testrand))./N;   //fréquence d'erreurs de 1ère espèce
//
printf('\nAvec alpha=%.3f et sur %d simulations, on observe\n...
une  fréquence empirique d''erreurs du premier type de\n...
%6.3f pour le test déterministe\n...
%6.3f pour le test randomisé',alpha,N,ftd,ftr)
```

Voici un exemple d'exécution.

```
-->;exec("/home/suquet/Enseignement/Scilab/Tests/tstsgnempH0.sce");

Avec alpha=0.050 et sur 5000 simulations, on observe
une  fréquence empirique d'erreurs du premier type de
 0.011 pour le test déterministe
 0.055 pour le test randomisé
```

Références

[1] D. Dacunha-Castelle et M. Duflo, *Probabilités et statistiques, 1. Problèmes à temps fixe.* Masson 1982.