

Algorithmes en probabilités

Illustrations avec XCAS et CAML

Guillaume CONNAN

IREM de Nantes

18 janvier 2010

Sommaire

- 1 Pile ou face
- 2 Tirage de boules
- 3 Problème du Duc de Toscane
- 4 Le lièvre et la tortue
- 5 Intervalle de fluctuation en 2^{nde}

Sommaire

- 1 Pile ou face
- 2 Tirage de boules
- 3 Problème du Duc de Toscane
- 4 Le lièvre et la tortue
- 5 Intervalle de fluctuation en 2nde

On lance trois fois de suite une pièce de monnaie. On veut compter combien de fois pile (ou face) tombe.

Avec XCAS, on utilisera :

- `rand(k)` qui renvoie un entier aléatoirement compris entre 0 et $k - 1$;
- `count_eq(entier, liste)` compte le nombre de fois ou entier apparaît dans `liste`

```
piece(n):={
T:=NULL;
for(k:=1;k<=n;k:=k+1){
X:=0;
if(rand(2)==1){X:=X+1}
if(rand(2)==1){X:=X+1}
if(rand(2)==1){X:=X+1}
T:=T,X;
}
return(evalf([count_eq(0,[T])/n,count_eq(1,[T])/n,count_eq
(2,[T])/n,count_eq(3,[T])/n]
))};;
```

```
piece(5000)
```

renvoie [0.1262,0.387,0.366,0.1208].

En effet, on peut vérifier avec `binomial(n,k,p)` qui renvoie

$$\binom{n}{k} p^k (1-p)^{n-k} :$$

```
[seq(binomial(3,k,0.5),k=0..3)]
```

renvoie [0.125,0.375,0.375,0.125]

```
piece(5000)
```

renvoie [0.1262,0.387,0.366,0.1208].

En effet, on peut vérifier avec `binomial(n,k,p)` qui renvoie

$\binom{n}{k} p^k (1-p)^{n-k}$:

```
[seq(binomial(3,k,0.5),k=0..3)]
```

renvoie [0.125,0.375,0.375,0.125]


```
piece(5000)
```

renvoie [0.1262,0.387,0.366,0.1208].

En effet, on peut vérifier avec `binomial(n,k,p)` qui renvoie

$$\binom{n}{k} p^k (1-p)^{n-k} :$$

```
[seq(binomial(3,k,0.5),k=0..3)]
```

renvoie [0.125,0.375,0.375,0.125]

Sommaire

1 Pile ou face

2 Tirage de boules

3 Problème du Duc de Toscane

4 Le lièvre et la tortue

5 Intervalle de fluctuation en 2nde

On dispose de trois urnes, la première contenant 7 boules blanches et 4 noires, la deuxième 5 blanches et 2 noires, la troisième 6 blanches et 3 noires.

On tire une boule dans chaque urne et on note le nombre de boules blanches obtenues.

On adaptera l'algorithme précédent.

```
boules(n) := {  
  T := NULL;  
  for(k := 1; k <= n; k := k + 1) {  
    X := 0;  
    if(rand(11) < 7) {X := X + 1}  
    if(rand(7) < 5) {X := X + 1}  
    if(rand(9) < 6) {X := X + 1}  
    T := T, X;  
  }  
  return(evalf([count_eq(0, [T])/n, count_eq(1, [T])/n, count_eq  
    (2, [T])/n, count_eq(3, [T])/n]  
  )});;
```

On obtient $[0.0358, 0.2253818182, 0.4453272727, 0.2934909091]$

Or, par exemple, $\frac{7}{11} \times \frac{5}{7} \times \frac{6}{9} \approx 0,303$ et $\frac{4}{11} \times \frac{2}{7} \times \frac{3}{9} \approx 0,034$.

On obtient $[0.0358, 0.2253818182, 0.4453272727, 0.2934909091]$

Or, par exemple, $\frac{7}{11} \times \frac{5}{7} \times \frac{6}{9} \approx 0,303$ et $\frac{4}{11} \times \frac{2}{7} \times \frac{3}{9} \approx 0,034$.

Sommaire

1 Pile ou face

2 Tirage de boules

3 Problème du Duc de Toscane

4 Le lièvre et la tortue

5 Intervalle de fluctuation en 2nde

Cosme II de Médicis (Florence 1590-1621), Duc de Toscane, fut le protecteur de l'illustre Galilée (né à Pise le 15 février 1564 et mort à Florence le 8 janvier 1642) son ancien précepteur. Profitant d'un moment de répit du savant entre l'écriture d'un théorème sur la chute des corps et la création de la lunette astronomique, le Grand Duc lui soumet le problème suivant : il a observé qu'en lançant trois dés cubiques et en faisant la somme des numéros des faces, on obtient plus souvent 10 que 9, alors qu'il y a autant de façons d'obtenir 9 que 10, à savoir six.

On pourra utiliser `ranm(lignes,colonnes,'expérience')` qui renvoie une matrice `lignes×colonnes` où chaque terme est un résultat de `expérience`.

```
toscane(n):={  
T:=ranm(1,n,'rand(6)+rand(6)+rand(6)+3');  
return("Obtenir 9 :"+count_eq(9,T)*100.0/n+"%", " Obtenir 10  
      :"+count_eq(10,T)*100.0/n+"%")  
};;
```

Alors, pour 100 000 tirages, on obtient

"Obtenir 9 :11.263%", " Obtenir 10 :12.535%"

```
toscane(n):={  
T:=ranm(1,n,'rand(6)+rand(6)+rand(6)+3');  
return("Obtenir 9 :"+count_eq(9,T)*100.0/n+"%", " Obtenir 10  
      :"+count_eq(10,T)*100.0/n+"%")  
};;
```

Alors, pour 100 000 tirages, on obtient

"Obtenir 9 :11.263%", " Obtenir 10 :12.535%"

Sommaire

1 Pile ou face

2 Tirage de boules

3 Problème du Duc de Toscane

4 **Le lièvre et la tortue**

5 Intervalle de fluctuation en 2nde

On rappelle le jeu bien connu : on dispose d'un dé cubique. On le lance. Si un six sort, le lièvre gagne sinon la tortue avance d'une case. La tortue a gagné lorsqu'elle a avancé six fois de suite.

Il est aisé de montrer que la probabilité que la tortue gagne $\mathbb{P}(T)$ est égale à $(\frac{5}{6})^6 \approx 0,335$.

```
jeutortue(n):={  
  local simul,L,T,face,lancer;  
  L:=0;T:=0;  
  pour simul de 1 jusque n faire  
    face:=rand(6)+1; lancer:=1;  
    tantque(face<6) et (lancer<6) faire  
      face:=rand(6)+1;  
      lancer:=lancer+1;  
    ftantque  
    si face==6  
      alors L:=L+1  
      sinon T:=T+1  
    fsi  
  fpour  
  
  return("Tortue :"+T*100.0/n+"% Lièvre : "+L*100.0/n+"%")  
};;
```

Alors on peut simuler 100 000 parties en entrant :

```
jeutortue(100000)
```

et on obtient par exemple :

```
Tortue :33.639% Lièvre : 66.361%
```

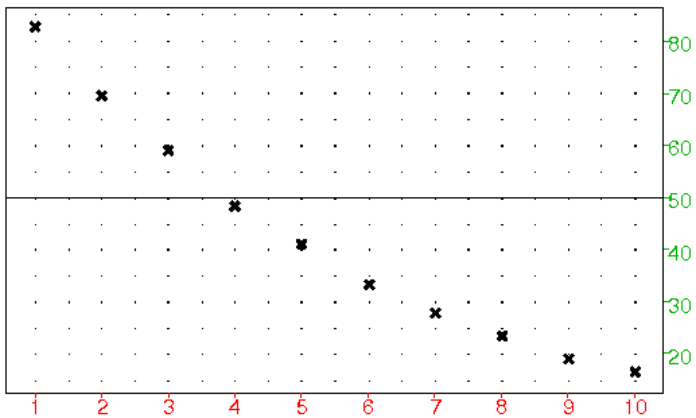
On peut modifier légèrement le programme précédent pour déterminer quel nombre maximum de lancers on doit fixer pour rendre le jeu favorable à la tortue.


```
jeutortue(n,lancer_max):={  
  local simul,L,T,face,lancer;  
  L:=0;T:=0;  
  pour simul de 1 jusque n faire  
    face:=rand(6)+1; lancer:=1;  
    tantque(face<6) et (lancer<lancer_max)faire  
      face:=rand(6)+1;  
      lancer:=lancer+1;  
    ftantque  
    si face==6  
      alors L:=L+1  
      sinon T:=T+1  
    fsi  
  fpour  
  
  return(T*100.0/n)  
};;
```

On n'a cette fois en sortie que le pourcentage de parties gagnées par la tortue.

On trace ensuite le nuage de points de coordonnées
(`lancer_max`, `jeutortue(10000,k)`) avec `k` variant de 1 à 10 par
exemple.

```
nuage_points([seq([k, jeutortue(10000,k)], k=1..10)], droite(  
  y=50))
```



Sommaire

1 Pile ou face

2 Tirage de boules

3 Problème du Duc de Toscane

4 Le lièvre et la tortue

5 Intervalle de fluctuation en 2^{nde}

Le document d'accompagnement affirme que pour des échantillons de taille n obtenus à partir d'un modèle de Bernoulli, 95% des mesures des fréquences mesurées sont comprises dans l'intervalle $\left[p - \frac{1}{\sqrt{n}}, p + \frac{1}{\sqrt{n}} \right]$ avec p la proportion à mesurer.

On simule ici 1000 échantillons de taille n d'un modèle de Bernoulli ayant une probabilité p exprimée en nombre entier de dixièmes.
`ranm(1,n,'rand(10)')` permet d'obtenir n nombres compris entre 0 et 9.
`count_inf(10*p,T)/n` permet de compter les occurrences dans T des nombres strictement inférieurs à $10p$: la fréquence théorique est p .
On compte 1 dès que la fréquence observée est dans l'intervalle de confiance.

Le document d'accompagnement affirme que pour des échantillons de taille n obtenus à partir d'un modèle de Bernoulli, 95% des mesures des fréquences mesurées sont comprises dans l'intervalle $\left[p - \frac{1}{\sqrt{n}}, p + \frac{1}{\sqrt{n}} \right]$ avec p la proportion à mesurer.

On simule ici 1000 échantillons de taille n d'un modèle de Bernoulli ayant une probabilité p exprimée en nombre entier de dixièmes.

`ranm(1, n, 'rand(10)')` permet d'obtenir n nombres compris entre 0 et 9.
`count_inf(10*p, T)/n` permet de compter les occurrences dans T des nombres strictement inférieurs à $10p$: la fréquence théorique est p .
On compte 1 dès que la fréquence observée est dans l'intervalle de confiance.

Le document d'accompagnement affirme que pour des échantillons de taille n obtenus à partir d'un modèle de Bernoulli, 95% des mesures des fréquences mesurées sont comprises dans l'intervalle $\left[p - \frac{1}{\sqrt{n}}, p + \frac{1}{\sqrt{n}} \right]$ avec p la proportion à mesurer.

On simule ici 1000 échantillons de taille n d'un modèle de Bernoulli ayant une probabilité p exprimée en nombre entier de dixièmes.

`ranm(1,n,'rand(10)')` permet d'obtenir n nombres compris entre 0 et 9.

`count_inf(10*p,T)/n` permet de compter les occurrences dans T des nombres strictement inférieurs à $10p$: la fréquence théorique est p .

On compte 1 dès que la fréquence observée est dans l'intervalle de confiance.

Le document d'accompagnement affirme que pour des échantillons de taille n obtenus à partir d'un modèle de Bernoulli, 95% des mesures des fréquences mesurées sont comprises dans l'intervalle $\left[p - \frac{1}{\sqrt{n}}, p + \frac{1}{\sqrt{n}} \right]$ avec p la proportion à mesurer.

On simule ici 1000 échantillons de taille n d'un modèle de Bernoulli ayant une probabilité p exprimée en nombre entier de dixièmes.

`ranm(1, n, 'rand(10)')` permet d'obtenir n nombres compris entre 0 et 9.
`count_inf(10*p, T)/n` permet de compter les occurrences dans T des nombres strictement inférieurs à $10p$: la fréquence théorique est p .

On compte 1 dès que la fréquence observée est dans l'intervalle de confiance.

Le document d'accompagnement affirme que pour des échantillons de taille n obtenus à partir d'un modèle de Bernoulli, 95% des mesures des fréquences mesurées sont comprises dans l'intervalle $\left[p - \frac{1}{\sqrt{n}}, p + \frac{1}{\sqrt{n}} \right]$ avec p la proportion à mesurer.

On simule ici 1000 échantillons de taille n d'un modèle de Bernoulli ayant une probabilité p exprimée en nombre entier de dixièmes.

`ranm(1, n, 'rand(10)')` permet d'obtenir n nombres compris entre 0 et 9.

`count_inf(10*p, T)/n` permet de compter les occurrences dans T des nombres strictement inférieurs à $10p$: la fréquence théorique est p .

On compte 1 dès que la fréquence observée est dans l'intervalle de confiance.

```
echantillon(n,p):={  
S:=0;  
  for(k:=1;k<=1000;k:=k+1){  
    T:=ranm(1,n,'rand(10)');  
    f:=count_inf(10*p,T)/n;  
    if((f>=(p-1./sqrt(n)))and(f<=(p+1./sqrt(n))))  
    then{S:=S+1}  
  }  
return(S*0.1)  
}::;
```

Par exemple, pour $n = 100$ et $p = 0,4$:

```
echantillon(100,0.4)
```

95.9

On relance autant de fois que l'on veut le calcul pour observer que le nombre obtenu est supérieur à 95.

Nous ne discuterons pas ici de la pertinence d'une telle expérience devant des élèves de 2^{nde}...

Par exemple, pour $n = 100$ et $p = 0,4$:

```
echantillon(100,0.4)
```

95.9

On relance autant de fois que l'on veut le calcul pour observer que le nombre obtenu est supérieur à 95.

Nous ne discuterons pas ici de la pertinence d'une telle expérience devant des élèves de 2^{nde}...

Par exemple, pour $n = 100$ et $p = 0,4$:

```
echantillon(100,0.4)
```

95.9

On relance autant de fois que l'on veut le calcul pour observer que le nombre obtenu est supérieur à 95.

Nous ne discuterons pas ici de la pertinence d'une telle expérience devant des élèves de 2^{nde}...