

Loi des grands nombres

Nicolas Gast, Arnaud Legrand

5 novembre 2015

- Simulation
 - Une petite fonction de simulation
 - Utilisation
 - Graphique
 - Temps de séjour dans le casino
 - État final
 - Étude d'un "grand" nombre de trajectoires
- Étude du cas fini
 - Cas où $p < 1/2$ (par exemple $p = 0.4$ pour commencer)
 - Cas où $p = 1/2$
 - Cas où $p > 1/2$ (par exemple $p = 0.6$ pour commencer)
 - Petit bilan
- Étude du cas infini.
 - Cas où $p < 1/2$
 - Cas où $p = 1/2$
 - Cas où $p > 1/2$
 - Petit bilan

Il commence à faire froid, les nuits sont longues, vous avez décidé de jouer au casino. Dans ce TD, on se propose de simuler un modèle simple de jeu de pile ou face et de déterminer par simulation quel est le temps moyen qu'un joueur joue avant de perdre toute sa mise ou d'arriver à gagner un montant M fixé à l'avance (les responsables de casinos sont souvent assez désagréables avec les petits malins qui gagnent trop d'argent...).

Le modèle est le suivant:

- Initialement, le joueur rentre au casino avec une somme fixée (par défaut, 10 euros)
- Tant que le joueur a $X \in \{1, \dots, M - 1\}$ euros, il mise un euro et gagne deux euros avec probabilité p (ce qui veut dire qu'au coup suivant, il a $X + 1$ euros avec probabilité p et $X - 1$ euros avec probabilité $1 - p$).

On cherche à évaluer le temps moyen que le joueur va jouer et son gain moyen dans les deux cas suivants:

- lorsque le joueur s'est fixé à l'avance une somme M et qu'il arrête de jouer lorsqu'il a M euros.
- lorsque le joueur joue tant qu'il n'est pas ruiné ($M = +\infty$)

Pour vous faire gagner du temps, la première section de ce document comporte quelques fonctions permettant de simuler ces différentes situations. Les sections suivantes sont à compléter par vos soins.

Simulation

Une petite fonction de simulation

Nous vous proposons d'utiliser la fonction suivante.

```
set.seed(42);

trajectory = function(init=10,gain_max=20,p=1/2,t_max=NA) {
  t=1;
  current_gain = init
  traj = c();
  traj[1] = current_gain;
  while (current_gain!=0 & (is.na(gain_max) | current_gain<gain_max) & (is.na
(t_max) | t<t_max)) {
    current_gain = current_gain + sample(x=c(-1,+1), size = 1, prob = c(p,1-p
));
    t = t + 1;
    traj[t] = current_gain;
  }
  return(traj);
}
```

Dans cette fonction, `init` représente l'état initial (ici, 10 euros en poche à l'entrée du casino), `gain_max` la somme au delà de laquelle le casino vous mettra dehors (ici 20 euros par défaut), et enfin `t_max` représente la durée maximale (éventuelle) pendant laquelle vous jouerez (ici, par défaut, il n'y a pas de limite!). Le paramètre `p` représente la probabilité de perte d'un euro et vaut par défaut 1/2. Notre fonction nous renvoie la trajectoire, c'est à dire le détail au cours du temps de nos gains.

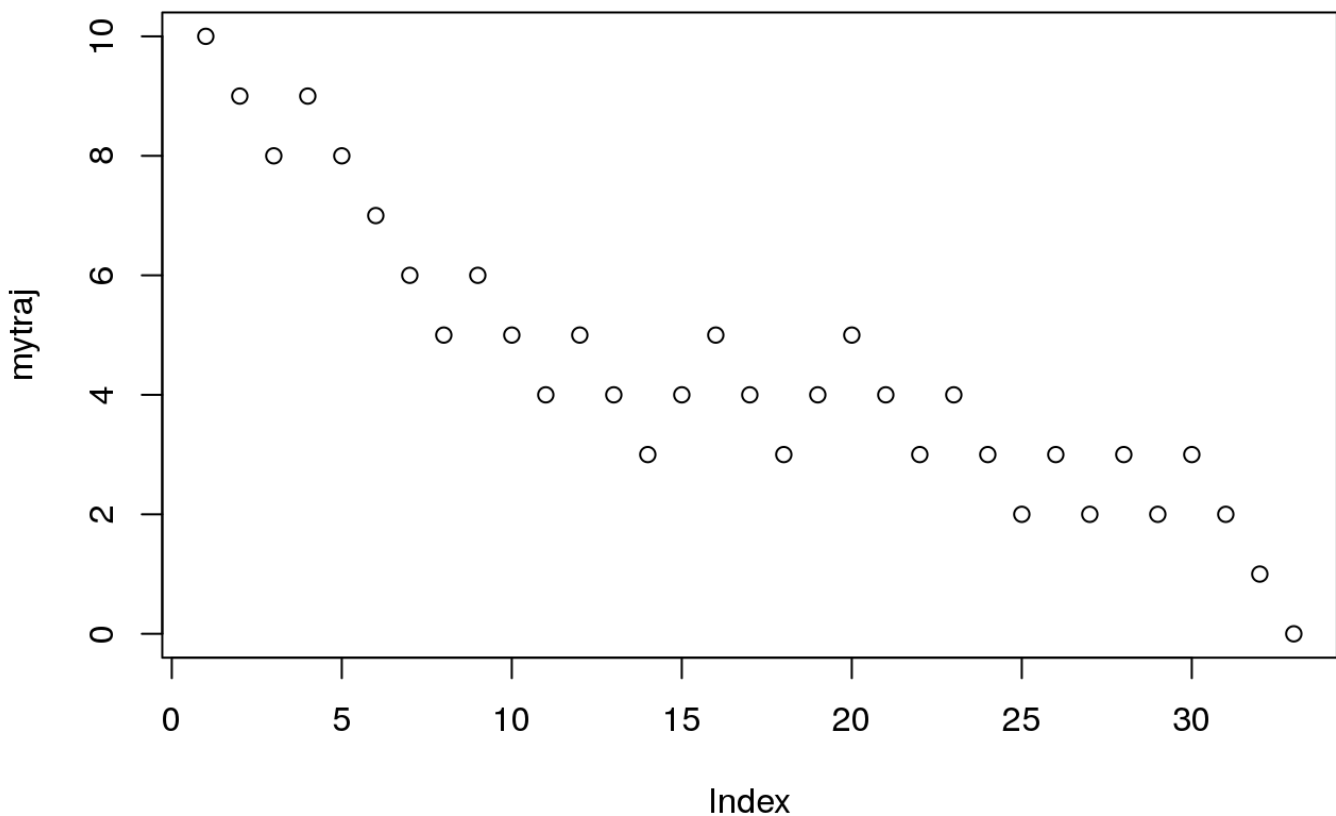
```
mytraj = trajectory();
```

Utilisation

Graphique

Puisqu'on récupère la trajectoire, il est aisé de regarder l'évolution de nos gains au cours du temps:

```
plot(mytraj);
```



Temps de séjour dans le casino

Le temps passé dans le casino correspond à la longueur de la trajectoire.

```
duration = function(traj) {
  length(traj);
}
duration(mytraj);
```

```
## [1] 33
```

État final

Le gain final correspond à la quantité d'argent que l'on a à la fin de la trajectoire.

```
final_gain = function(traj) {
  traj[length(traj)];
}
final_gain(mytraj);
```

[1] 0

Étude d'un “grand” nombre de trajectoires

Dans le cas où on veut pouvoir faire aisément varier le paramètre `p`, `init`, ou `gain_max`, il sera très pratique de “propager” ces paramètres à la fonction `trajectory` sans avoir besoin de les spécifier tous. Pour cela, on peut utiliser la notation `...` de R qui est bien pratique. Par exemple:

```
bankrupt_proba = function(N=1000, ...) {
  count = 0;
  for(i in 1:N) {
    mytraj = trajectory(...);
    if(final_gain(mytraj)==0) { count = count + 1; }
  }
  count/N;
}
```

Cette fonction réalise N (par défaut 1000) trajectoires et compte le nombre de fois où vous avez fait faillite:

```
bankrupt_proba();
```

[1] 0.513

Maintenant à vous de remplacer les “À vous de jouer!” par quelque chose d’intéressant. :)

Étude du cas fini

On suppose dans cette partie que le gain maximum autorisé par le casino est fini (par exemple 20 euros) et qu’au delà de cette somme, le casino vous mettra dehors. Vous étudierez pour chacun des cas suivants la probabilité de banqueroute, le temps passé au casino, et le gain moyen. Si vous avez un peu de temps, vous pourrez aussi étudier l’influence du point de départ `init`.

Cas où $p < 1/2$ (par exemple $p = 0.4$ pour commencer)

À vous de jouer!

Cas où $p = 1/2$

À vous de jouer!

Cas où $p > 1/2$ (par exemple $p = 0.6$ pour commencer)

À vous de jouer!

Petit bilan

À vous de jouer!

p	$< .5$	$= .5$	$> .5$
Proba banqueroute			
Temps passé			
Gain moyen			

Étude du cas infini.

On s'intéresse maintenant au cas infini (i.e., $\text{gain_max}=\text{NA}$). Il est alors possible que la simulation ne s'arrête pas. Si vous avez l'impression d'avoir des problèmes de durée de simulation un peu trop grands, vous prendrez donc soin de borner la durée de la simulation (e.g., $t_{\text{max}} = 1000$).

Encore une fois, vous vous intéresserez à la probabilité de banqueroute, au temps passé au casino, et au gain moyen (si cela a du sens...).

Afin de bien comprendre ce qu'il se passe, nous vous suggérons de réaliser ces études pour différentes valeurs de t_{max} (par exemple 10, 100, 1000) et de regarder graphiquement comment la probabilité de banqueroute et le temps passé évoluent avec t_{max} .

Cas où $p < 1/2$

À vous de jouer!

Cas où $p = 1/2$

À vous de jouer!

Cas où $p > 1/2$

À vous de jouer!

Petit bilan

À vous de jouer!

p	<.5	=.5	>.5
Proba banqueroute			
Temps passé			
Gain moyen			