

# RICM4: Probabilité et Simulation. DM1: Pierre Feuille Ciseaux

Arnaud Legrand

## CONTENTS

<b>I</b>	<b>Pierre, Feuille, Ciseau chez les poissons!</b>	1
I-A	Mise en oeuvre d'une petite simulation . . . . .	1
I-B	Le joueur biaisé . . . . .	2
I-C	Le joueur non biaisé . . . . .	4
<b>II</b>	<b>Apprentissage</b>	5
II-A	Un algorithme qui apprend et joue optimalement . . . . .	5
<b>III</b>	<b>Bonus au choix</b>	8
III-A	Extension spéciale Big Bang Theory . . . . .	8
III-B	Proposez une stratégie non aléatoire . . . . .	8

### I. PIERRE, FEUILLE, CISEAU CHEZ LES POISSONS!

#### A. Mise en oeuvre d'une petite simulation

Commençons par une petite fonction générant une série d'actions sans mémoire et suivant une loi de probabilité donnée.

```
PFC = as.factor(c("Pierre", "Feuille", "Ciseau"));
PFC_sequence = function(N=1000, P=c(1/3, 1/3, 1/3)) {
  sample(PFC, N, P, replace=T)
}
PFC_sequence(20)
```

```
[1] Pierre Ciseau Feuille Pierre Feuille Pierre Pierre Feuille Feuille
[10] Ciseau Feuille Pierre Feuille Ciseau Pierre Pierre Pierre Ciseau
[19] Pierre Ciseau
Levels: Ciseau Feuille Pierre
```

Vérifions que le code est "correct" et que ça n'a pas l'air trop biaisé.

```
summary(PFC_sequence(1000))
```

```
Ciseau Feuille Pierre
 330     326     344
```

Bon, comment coder ces comparaisons Pierre, Feuille, Ciseau "simplement". Je les ai représentées comme des facteurs mais je peux convertir ces facteurs en numéros et à partir de là, utiliser l'ordre sur ces nombres modulo 3.

```
PFC
```

```
as.numeric(PFC)
```

```
[1] Pierre Feuille Ciseau
Levels: Ciseau Feuille Pierre
[1] 3 2 1
```

C'est l'ordre alphabétique qui a été utilisé. On a donc avec ce codage "3 < 2 < 1 < 3". Comme demandé dans l'énoncé, j'utiliserai le point de vue du joueur B pour déterminer les gains:

```
PFC_outcome = function(A_seq, B_seq) {
  ## print(A_seq);
  ## print(B_seq);
  A_seq = as.numeric(A_seq);
  B_seq = as.numeric(B_seq);
```

```

outcome = ((3+A_seq-B_seq)%%3)
outcome = ifelse(outcome==2,-1,outcome)
## print(outcome)
outcome
}
A = PFC_sequence(8)
B = PFC_sequence(8)
A
B
PFC_outcome(A,B)

[1] Feuille Pierre Pierre Pierre Ciseau Ciseau Feuille Pierre
Levels: Ciseau Feuille Pierre
[1] Ciseau Feuille Feuille Ciseau Ciseau Feuille Ciseau Feuille
Levels: Ciseau Feuille Pierre
[1] 1 1 1 -1 0 -1 1 1

```

Écrivons une petite fonction calculant le gain moyen d'une partie entre deux joueurs sans mémoire:

```

PFC_memoryless_game = function(N,pA,pB) {
  A = PFC_sequence(N,pA)
  B = PFC_sequence(N,pB)
  mean(PFC_outcome(A,B))
}

```

### B. Le joueur biaisé

On suppose que le joueur  $A$  joue avec probabilités  $P^{(A)} = (1/4, 1/4, 1/2)$ .

```

pA = c(1/4,1/4,1/2)
PFC_memoryless_game(1E6,pA,c(1/4,1/4,1/2))
PFC_memoryless_game(1E6,pA,c(1/3,1/3,1/3))

[1] -0.001121
[1] 0.000253

```

Mmmh, pas fameux. Quand je répète le bloc précédent, ça oscille autour de 0. J'ai choisi un million de tirages pour que ça n'oscille "pas trop". Si on avait vu les cours sur l'intervalle de confiance, on saurait qu'il y a 99.9% de chances pour que ça correspond à une approximation 0.003 près ( $3 * 1/\sqrt{(N)}$ ), Autrement, dit, les deux premières décimales sont assez fiables. Tu parles d'un jeu inintéressant...

Regardons s'il n'y a pas une stratégie plus intéressante

```

pA = c(1/4,1/4,1/2)
Gain = data.frame();
for(x in ((0:10)/10)) {
  for(y in ((0:10)/10)) {
    if(x+y<=1) {
      gain = PFC_memoryless_game(1000000,pA,c(x,y,1-(x+y)));
      Gain = rbind(Gain, data.frame(x=x,y=y,gain=gain));
    }
  }
}
head(Gain);
tail(Gain);
summary(Gain);

  x  y  gain
1 0 0.0 -0.000015
2 0 0.1 -0.024234
3 0 0.2 -0.050794
4 0 0.3 -0.076173
5 0 0.4 -0.100791
6 0 0.5 -0.124812

```

```

  x   y   gain
61 0.8 0.0 0.200174
62 0.8 0.1 0.174111
63 0.8 0.2 0.150387
64 0.9 0.0 0.224514
65 0.9 0.1 0.199667
66 1.0 0.0 0.249588

```

```

      x              y              gain
Min.   :0.0000   Min.   :0.0000   Min.   :-2.485e-01
1st Qu.:0.1000   1st Qu.:0.1000   1st Qu.: -7.591e-02
Median :0.3000   Median :0.3000   Median :-2.100e-05
Mean   :0.3333   Mean   :0.3333   Mean    : 5.461e-05
3rd Qu.:0.5000   3rd Qu.:0.5000   3rd Qu.: 7.578e-02
Max.   :1.0000   Max.   :1.0000   Max.    : 2.496e-01

```

Le max est à 0.2496. Vers où se trouve-t-il ?

```
head(Gain[order(-Gain$gain),])
```

```

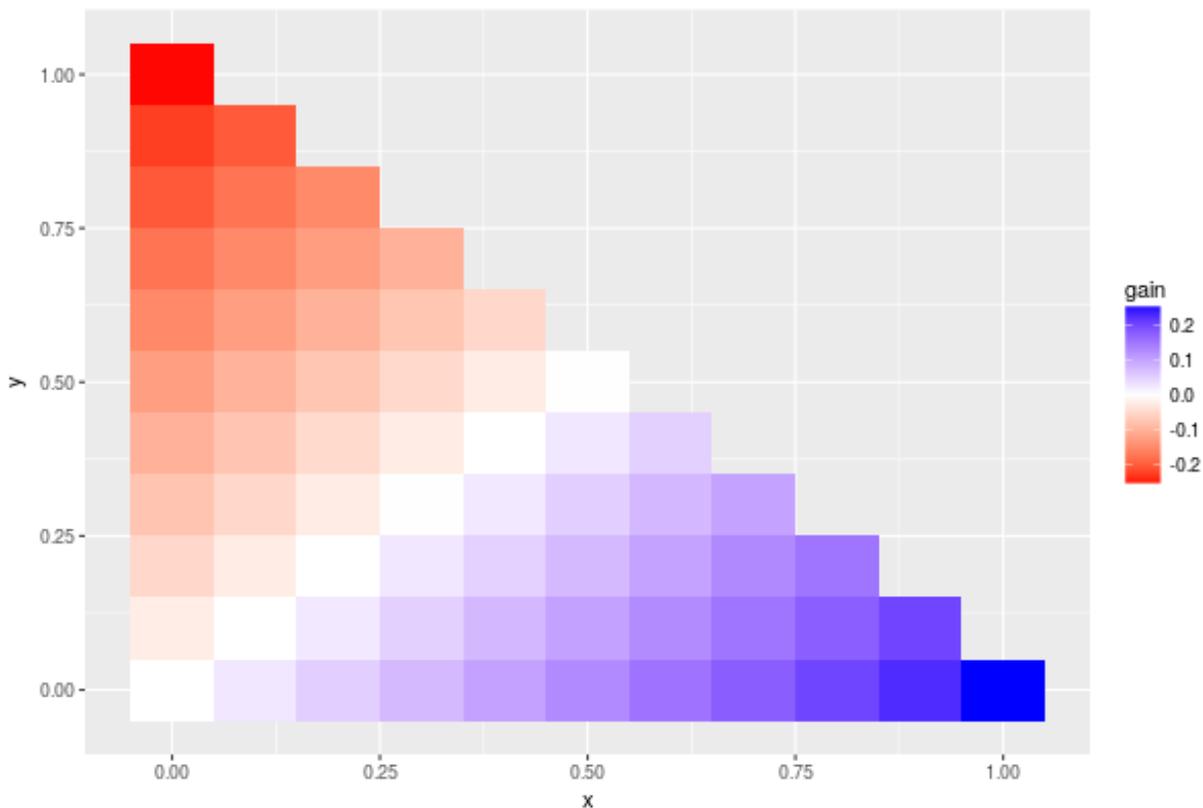
  x   y   gain
66 1.0 0.0 0.249588
64 0.9 0.0 0.224514
61 0.8 0.0 0.200174
65 0.9 0.1 0.199667
57 0.7 0.0 0.174610
62 0.8 0.1 0.174111

```

C'est à peu près clair, il faut jouer systématiquement là où ça fait mal. Il joue souvent "Ciseau", il faut jouer "Pierre" systématiquement pour écraser A. Regardons si d'autres stratégies sont bonnes:

```
library(ggplot2)
```

```
ggplot(Gain, aes(x=x,y=y,fill=gain)) + geom_tile() + scale_fill_gradient2(
  low = "red", high = "blue",
  mid = "white")
```



C'est super linéaire comme comportement. Il doit y avoir une formule toute simple. :)

On a le tableau suivant:

$A \backslash B$	$P(x)$	$F(y)$	$C(1 - (x + y))$
$P(1/4)$	0	1	-1
$F(1/4)$	-1	0	1
$C(1/2)$	1	-1	0

L'espérance du gain de  $B$  est donc  $1/4.(y - (1 - (x + y))) + 1/4.(-x + (1 - (x + y))) + 1/2.(x - y) = (x - y)/4$ . C'est bien maximum pour  $(x, y) = (1, 0)$ !

### C. Le joueur non biaisé

Supposons maintenant que le joueur  $A$  est non biaisé, i.e., que  $P^{(A)} = (1/3, 1/3, 1/3)$ .

```
pA = c(1/3, 1/3, 1/3)
```

```
Gain = data.frame();
```

```
for(x in ((0:10)/10)) {
```

```
  for(y in ((0:10)/10)) {
```

```
    if(x+y<=1) {
```

```
      gain = PFC_memoryless_game(1000000, pA, c(x, y, 1 - (x + y)));
```

```
      Gain = rbind(Gain, data.frame(x=x, y=y, gain=gain));
```

```
    }
```

```
  }
```

```
}
```

```
summary(Gain)
```

x	y	gain
Min. :0.0000	Min. :0.0000	Min. :-1.905e-03
1st Qu.:0.1000	1st Qu.:0.1000	1st Qu.: -7.897e-04
Median :0.3000	Median :0.3000	Median : -1.130e-04
Mean :0.3333	Mean :0.3333	Mean : -9.118e-05
3rd Qu.:0.5000	3rd Qu.:0.5000	3rd Qu.: 5.462e-04
Max. :1.0000	Max. :1.0000	Max. : 2.154e-03

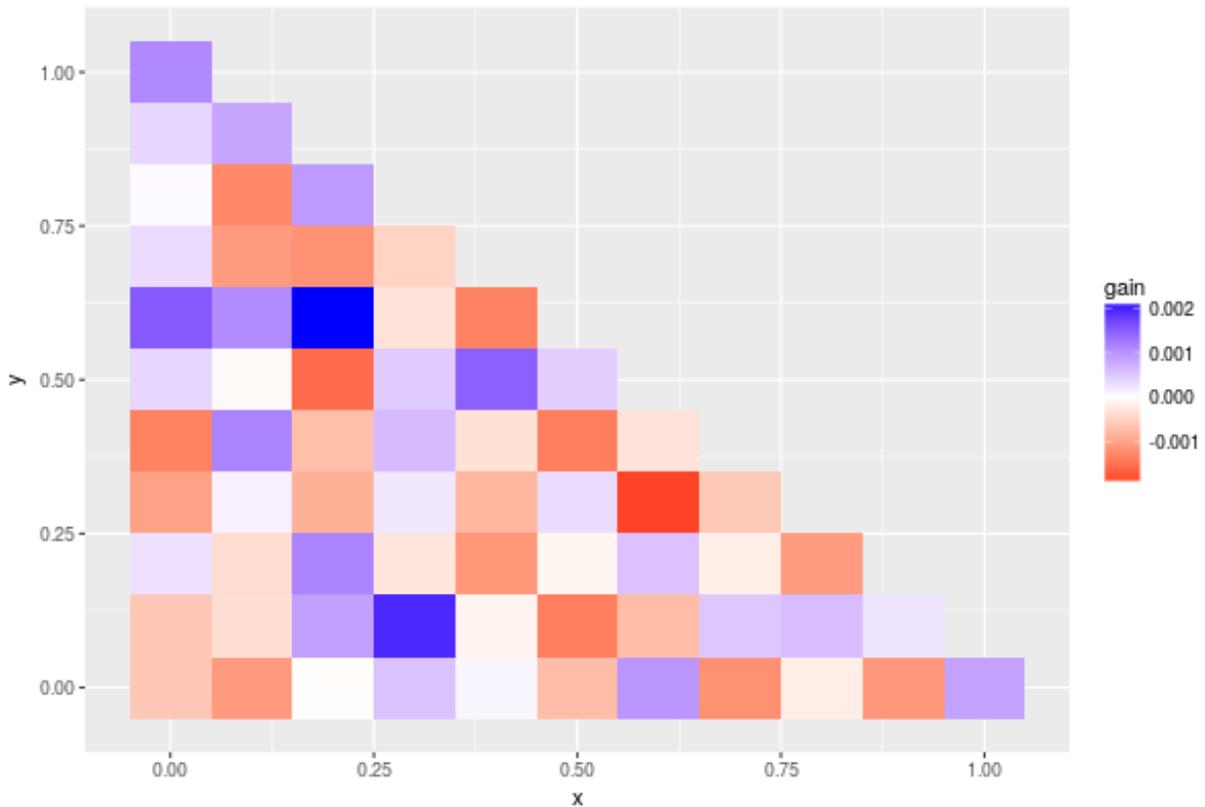
Les gains pour  $B$  sont partout proche de 0. On dirait que  $A$  ne peut pas perdre! Ceci dit, il ne peut pas gagner non plus. Regardons, si on voit une structure ou pas:

```
library(ggplot2)
```

```
ggplot(Gain, aes(x=x, y=y, fill=gain)) + geom_tile() + scale_fill_gradient2(
```

```
  low = "red", high = "blue",
```

```
  mid = "white")
```



Et non, ça a l'air complètement bruité et proche de 0. Si on reprend le calcul précédent, on verra vite que tout se simplifie et que l'espérance du gain est bien nulle.

## II. APPRENTISSAGE

Notre objectif est maintenant de créer un programme qui arrive à gagner contre un joueur humain. Nous savons que les humains jouent mal au hasard et peuvent en particulier avoir du mal à maintenir l'uniformité.

### A. Un algorithme qui apprend et joue optimalement

Si notre adversaire est biaisé et n'arrive pas à maintenir l'uniformité, il suffit de jouer systématiquement contre ce qu'il joue le plus fréquemment.

```
PFC_Strat1 = fonction(Aseq) {
  history = PFC; # Let's assume A played both three values before at least once
  Bseq = rep(PFC[1], length(Aseq));
  for(i in (1:length(Aseq))) {
    ## print("*****")
    ## print(history);
    stat = c(length(history[history=="Pierre"]),
              length(history[history=="Feuille"]),
              length(history[history=="Ciseau"]))
    maxplayed = head(PFC[stat == max(stat)], n=1)
    ## print(maxplayed)
    Bseq[i] = PFC[as.numeric(PFC) == (((as.numeric(maxplayed)+1)%%3)+1)]
    ## print("-----")
    ## print(Bseq);
    history[length(history)+1] = Aseq[i];
  }
  Bseq
}
```

Regardons s'il gagne contre un joueur biaisé:

```
pA = c(1/4, 1/4, 1/2)
A_seq = PFC_sequence(200, pA)
```

```

B_seq = PFC_Strat1(A_seq)
A_seq
B_seq
mean(PFC_outcome(A_seq,B_seq))

 [1] Ciseau Feuille Pierre Pierre Feuille Ciseau Ciseau Ciseau Ciseau
[10] Pierre Pierre Ciseau Ciseau Pierre Pierre Ciseau Ciseau Pierre
[19] Pierre Feuille Ciseau Ciseau Ciseau Pierre Ciseau Ciseau Ciseau
[28] Ciseau Ciseau Ciseau Ciseau Ciseau Ciseau Feuille Ciseau Ciseau
[37] Pierre Pierre Ciseau Feuille Ciseau Ciseau Ciseau Pierre Pierre
[46] Pierre Pierre Feuille Ciseau Feuille Pierre Ciseau Ciseau Ciseau
[55] Ciseau Pierre Pierre Ciseau Ciseau Pierre Pierre Ciseau Ciseau
[64] Feuille Pierre Pierre Pierre Ciseau Ciseau Ciseau Ciseau Feuille
[73] Pierre Ciseau Ciseau Ciseau Pierre Pierre Ciseau Pierre Ciseau
[82] Feuille Ciseau Pierre Ciseau Feuille Pierre Feuille Ciseau Ciseau
[91] Pierre Pierre Ciseau Ciseau Ciseau Ciseau Ciseau Pierre Feuille
[100] Ciseau Pierre Ciseau Ciseau Feuille Ciseau Pierre Ciseau Ciseau
[109] Feuille Pierre Pierre Ciseau Ciseau Pierre Feuille Ciseau Feuille
[118] Ciseau Feuille Feuille Ciseau Ciseau Ciseau Ciseau Pierre Ciseau
[127] Ciseau Ciseau Feuille Feuille Feuille Pierre Ciseau Feuille Ciseau
[136] Ciseau Pierre Pierre Pierre Feuille Feuille Feuille Feuille Ciseau
[145] Pierre Ciseau Pierre Ciseau Feuille Feuille Pierre Pierre Feuille
[154] Ciseau Ciseau Feuille Ciseau Ciseau Ciseau Ciseau Feuille Ciseau Feuille
[163] Ciseau Ciseau Ciseau Ciseau Ciseau Ciseau Pierre Feuille Feuille
[172] Ciseau Pierre Feuille Feuille Ciseau Pierre Feuille Ciseau Ciseau
[181] Feuille Pierre Ciseau Ciseau Ciseau Feuille Ciseau Ciseau Feuille
[190] Ciseau Ciseau Ciseau Pierre Pierre Feuille Ciseau Ciseau Pierre
[199] Ciseau Ciseau

Levels: Ciseau Feuille Pierre
 [1] Feuille Pierre Ciseau Feuille Feuille Feuille Feuille Pierre Pierre
[10] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[19] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[28] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[37] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[46] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[55] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[64] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[73] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[82] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[91] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[100] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[109] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[118] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[127] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[136] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[145] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[154] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[163] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[172] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[181] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[190] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
[199] Pierre Pierre

Levels: Ciseau Feuille Pierre
[1] 0.29

```

Alors évidemment, cet algorithme joue optimalement mais un humain pas trop bête devrait vite comprendre cette stratégie et arriver à rééquilibrer, par exemple en jouant systématiquement contre ce que l'ordinateur vient de jouer...

Pour que cette stratégie soit un peu moins grossièrement détectable, jouons un peu au hasard quand même (avec

une proba 1/2 par exemple).

```
PFC_Strat2 = function(Aseq, proba = 1/2) {
  history = PFC; # Let's assume A played both three values before at least once
  Bseq = rep(PFC[1], length(Aseq));
  for(i in (1:length(Aseq))) {
    if(runif(1)<proba) {
      stat = c(length(history[history=="Pierre"]),
               length(history[history=="Feuille"]),
               length(history[history=="Ciseau"]))
      maxplayed = head(PFC[stat == max(stat)], n=1)
      Bseq[i] = PFC[as.numeric(PFC)==(((as.numeric(maxplayed)+1)%%3)+1)]
    } else {
      Bseq[i] = sample(PFC, 1)
    }
    history[length(history)+1] = Aseq[i];
  }
  Bseq
}
```

Regardons s'il gagne contre un joueur biaisé:

```
pA = c(1/4, 1/4, 1/2)
A_seq = PFC_sequence(200, pA)
B_seq = PFC_Strat2(A_seq)
A_seq
B_seq
mean(PFC_outcome(A_seq, B_seq))

 [1] Pierre Pierre Ciseau Feuille Pierre Feuille Ciseau Feuille Feuille
[10] Feuille Feuille Ciseau Ciseau Feuille Ciseau Ciseau Ciseau Feuille
[19] Ciseau Ciseau Pierre Ciseau Pierre Ciseau Feuille Ciseau Pierre
[28] Feuille Pierre Pierre Ciseau Pierre Feuille Feuille Ciseau Ciseau
[37] Ciseau Pierre Pierre Ciseau Ciseau Ciseau Pierre Ciseau Feuille
[46] Ciseau Feuille Ciseau Ciseau Pierre Ciseau Feuille Ciseau Feuille
[55] Feuille Pierre Pierre Feuille Feuille Pierre Feuille Ciseau Feuille
[64] Ciseau Feuille Feuille Ciseau Ciseau Ciseau Ciseau Ciseau Ciseau
[73] Pierre Pierre Feuille Pierre Pierre Feuille Pierre Pierre Feuille
[82] Ciseau Pierre Pierre Pierre Ciseau Feuille Ciseau Ciseau Feuille
[91] Ciseau Feuille Ciseau Feuille Pierre Feuille Pierre Feuille Pierre
[100] Feuille Feuille Pierre Ciseau Ciseau Pierre Pierre Ciseau Ciseau
[109] Pierre Pierre Ciseau Ciseau Ciseau Ciseau Pierre Ciseau Ciseau
[118] Ciseau Ciseau Feuille Ciseau Ciseau Ciseau Ciseau Feuille Pierre
[127] Ciseau Ciseau Pierre Ciseau Ciseau Pierre Ciseau Ciseau Pierre
[136] Ciseau Ciseau Ciseau Feuille Feuille Ciseau Ciseau Feuille Ciseau
[145] Pierre Feuille Ciseau Ciseau Pierre Feuille Pierre Ciseau Ciseau
[154] Ciseau Pierre Feuille Pierre Pierre Feuille Ciseau Pierre Ciseau
[163] Feuille Ciseau Feuille Pierre Feuille Pierre Pierre Pierre Ciseau
[172] Ciseau Pierre Pierre Pierre Feuille Feuille Ciseau Pierre Ciseau
[181] Pierre Pierre Ciseau Feuille Feuille Ciseau Ciseau Ciseau Ciseau
[190] Ciseau Ciseau Ciseau Pierre Ciseau Feuille Pierre Pierre Feuille
[199] Ciseau Ciseau

Levels: Ciseau Feuille Pierre
 [1] Feuille Ciseau Feuille Pierre Feuille Pierre Feuille Ciseau Feuille
[10] Ciseau Ciseau Ciseau Ciseau Ciseau Ciseau Ciseau Feuille Ciseau
[19] Ciseau Ciseau Ciseau Pierre Feuille Feuille Feuille Pierre Pierre
[28] Pierre Feuille Ciseau Ciseau Pierre Feuille Pierre Ciseau Pierre
[37] Pierre Feuille Pierre Feuille Pierre Pierre Pierre Pierre Pierre
[46] Pierre Feuille Pierre Pierre Pierre Pierre Ciseau Pierre Pierre
[55] Feuille Feuille Feuille Pierre Pierre Ciseau Pierre Pierre Pierre
[64] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre
```

```

[73] Feuille Pierre Pierre Pierre Ciseau Pierre Pierre Pierre Pierre
[82] Feuille Pierre Ciseau Pierre Pierre Pierre Pierre Pierre Pierre
[91] Ciseau Pierre Pierre Feuille Feuille Pierre Pierre Feuille Feuille
[100] Pierre Ciseau Pierre Ciseau Ciseau Feuille Pierre Ciseau Feuille
[109] Ciseau Pierre Pierre Ciseau Pierre Pierre Pierre Pierre Pierre
[118] Feuille Pierre Ciseau Pierre Pierre Feuille Ciseau Pierre Pierre
[127] Feuille Ciseau Pierre Pierre Feuille Pierre Pierre Feuille Pierre
[136] Pierre Pierre Feuille Ciseau Feuille Feuille Pierre Pierre Ciseau
[145] Pierre Pierre Feuille Pierre Feuille Ciseau Pierre Pierre Pierre
[154] Pierre Feuille Pierre Ciseau Feuille Feuille Pierre Pierre Pierre
[163] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Pierre Feuille
[172] Pierre Pierre Ciseau Pierre Ciseau Feuille Ciseau Feuille Pierre
[181] Pierre Pierre Pierre Pierre Feuille Pierre Pierre Feuille Feuille
[190] Pierre Pierre Pierre Pierre Pierre Pierre Pierre Feuille Pierre
[199] Ciseau Pierre
Levels: Ciseau Feuille Pierre
[1] 0.095

```

Bon, on gagne mais 1/2, ça reste un peu trop visible, mais il faut savoir gagner peu pour gagner. Avec .1, on n’y verra que du feu! :) Quoi qu’il en soit, on a un compromis.

```

pA = c(1/4, 1/4, 1/2)
A_seq = PFC_sequence(2000, pA)
B_seq = PFC_Strat2(A_seq, proba=.1)
mean(PFC_outcome(A_seq, B_seq))
[1] 0.0545

```

### III. BONUS AU CHOIX

#### A. Extension spéciale Big Bang Theory

Adapter votre programme et votre étude au jeu Pierre-Papier-Ciseaux-Lézard-Spock.

Le plus dur est d’arriver à trouver un codage des facteurs avec lequel les calculs de modulo marchent bien.

#### B. Proposez une stratégie non aléatoire

Je vous invite à ce document (un peu long, mais rigolo) sur le sujet: <http://apiacoa.org/publications/2004/rossi2004opponent.pdf>