Statistical analysis of the Stigmer data

Contents

1 Importation of the data 1
1.1 Detection of anomalies 1

2 Statistical analysis of Stigmer 15 4
2.1 Statistical distribution of the gain per step 4
2.2 Analysis under R0 5
2.3 Analysis under R1 13
2.4 Analysis under R2 26

3 Statistical analysis of Stigmer 50 35

Packages needed:

```
require(tidyverse)
require(rpart)
require(rpart.plot)
```

Vocabulary used:

- there are three **rules** : R0, R1 and R2,
- a session is constituted of 15 **rounds**,
- Each round is constituted of 5 **steps**,
- The **gain per step** is equal to 0, 15 (or 50), 99,
- The **gain per round** is the sum of the 5 gains obtained during a round,
- The **final gain** is the sum of the gains obtained by a player during a session,
- A **cell** is one of the 9 cells of the grid,
- Under R1, a player leaves a **coin** in each cell visited,
- Under R2, a player has the choice to leave or not a coin in the visited cell.

1 Importation of the data

The function *import_files()* (codes presented in .Rmd file) leads us to import the data depending on the intermediate value in the game (15 or 50).

To import the data, we call the previous function:

```
file_15 <- import_files(15)
file_50 <- import_files(50)
```

1.1 Detection of anomalies

These anomalies have been detected in the version of the data given on March the 5th 2018

1.1.1 Stigmer 15

- In: Rule 1, session 03, in round "7B", a player is called 4B instead of B4. We have corrected this anomaly:

```r
file_15[file_15$player == "4B", "player"] <- "B4"
```

- In: Rule 2, session 01, we found two rounds with name “14A” : we kept only one round (the one with the earlier date).
- In: Rule 2, session 03, there is one missing row in round “11A”. A player did not play at step 1. We impute this missing value by 0, considering that this was the value he was most likely to get. We have corrected this:

```r
file_15[is.na(file_15$gain_1), c("gain_1", "action_1", "gain")] <-
    c(0, 0, 114)
file_15[is.na(file_15$gain_1), "cell_1"] <- "1_1"
```

- In: Rule 2, session 01bis, in round “14”, a player is called “A1” instead of “B4”. We have corrected this :

```r
file_15[file_15$player == "A1" & file_15$session == "session_01Bis" &
        file_15$rule == "R2" & file_15$round == "T14" & file_15$gain_2 == 0 , "player"] <- "B4"
```

- In: Rule 2, session 01bis, in round “15”, a player is called “B4” instead of “A1”. We have corrected this :

```r
file_15[file_15$player == "B4" & file_15$session == "session_01Bis" &
        file_15$rule == "R2" & file_15$round == "T15" & file_15$gain_2 == 0 , "player"] <- "A1"
```

- In Rule 1, players should have left a coin at each step. We notice that this is not necessarily the case :
    - player A2 in session 01, round “1A”, “3A”, “5A”, “12A”, “13A”,
    - player B1 in session 02, round “6B”, “7B”, “8B”, “12B”,
    - player B2 in session 01, round “6B”,
    - player A2 in session 02, round “11A”,
    - player A1 in session 03, round “1A”,
    - player B4 in session 03, round “6B”.

Remark: no corrections have been applied because in Rule 1, we implicitly suppose that all players have indicated their choices.

- We now identify the players who did not play value 99 although they knew where it was located. For doing this, we created the function *detect_bad_player()* (codes presented in .Rmd file) :

We print the bad players :

```r
detec_bad_player(file_15)
```

```
## ** Bad players in R0
## In session_03 , player A5 has played:
##     round gain_1 gain_2 gain_3 gain_4 gain_5
## 155   T01      0     99     15      0      0
## In session_03 , player B3 has played:
##     round gain_1 gain_2 gain_3 gain_4 gain_5
## 158   T01      0      0     15     99     15
## In session_03 , player B4 has played:
##     round gain_1 gain_2 gain_3 gain_4 gain_5
## 179   T03      0     15     15     99      0
## 209   T06     15     99     15      0      0
## 229   T08     15      0     99      0     15
## 249   T10      0     15     99      0      0
## 259   T11      0     15     15     99      0
## 279   T13     99     15      0      0      0
## In session_03 , player B5 has played:
##     round gain_1 gain_2 gain_3 gain_4 gain_5
## 230   T08     99     99     99     99      0
##
## ** Bad players in R1
## In session_01 , player A3 has played:
##     round gain_1 gain_2 gain_3 gain_4 gain_5
## 3     T01      0      0     99     15      0
## 13    T02      0      0     99      0     15
## 23    T03      0     15     15     99     15
## 33    T04     99     15      0      0      0
## 53    T06     15      0     99      0      0
## 63    T07     15     99      0      0      0
## 73    T08     15     15      0     99      0
## 83    T09      0      0     99     15      0
## 103   T11      0      0     99     15     15
## 113   T12     15     99     15      0     15
## 133   T14     15      0     99      0      0
## 143   T15      0      0     15     99     15
## In session_03 , player B4 has played:
##     round gain_1 gain_2 gain_3 gain_4 gain_5
## 319   T02      0     99      0      0     15
## 366   T07      0     15     15     99      0
## 389   T09      0      0     99     15      0
## 429   T13      0     15     99      0     15
## 449   T15     99     15      0     15      0
##
## ** Bad players in R2
## In session_01 , player A3 has played:
##   round gain_1 gain_2 gain_3 gain_4 gain_5
## 3   T01     15      0     99      0      0
## In session_03 , player B4 has played:
##     round gain_1 gain_2 gain_3 gain_4 gain_5
## 479   T03      0     99      0      0     15
## 489   T04      0     15      0     99      0
## 499   T05      0     99      0     15     15
## 559   T11     99      0      0     15      0
## 569   T12      0      0     99     15      0
## 599   T15     15     99      0      0     15
```

Remark: we did not apply any corrections for these individuals and kept them in the analysis.

1.1.2 Stigmer 50

- In : Rule 0 / session 01 / round “3B”, player B4 did not play at all during this game. Correction done: none.
- In Rule 1, players should have indicated all their choices (value 1). We noticed that this is not necessarily the case :
  - player B2 in session 01, round “1B”,
  - player A3 in session 01, round “3A”, “5A”,
  - player A4 in session 01, round “1A”,
  - player A3 in session 01, round “2A”, “3A”, “4A”.

Remark: no corrections have been applied because in Rule 1, we implicitly suppose that all players have indicated their choices.

- We will now identify the players who did not play value 99 although they knew where it was located :

```
detec_bad_player(file_50)
```

```
## ** Bad players in R0
## In session_01 , player A1 has played:
##    round gain_1 gain_2 gain_3 gain_4 gain_5
## 30   T04     99      0     50     99     99
## 40   T05     50      0     99      0      0
## 60   T07      0      0     99      0      0
## In session_02 , player A2 has played:
##     round gain_1 gain_2 gain_3 gain_4 gain_5
## 171   T03      0      0     99     50     50
## 181   T04     50     50     99      0      0
## 201   T06     50      0     99     50     50
## 261   T12     99     50      0      0      0
## 271   T13      0      0      0     99     50
## 291   T15      0     50     99      0     50
## In session_02 , player B3 has played:
##     round gain_1 gain_2 gain_3 gain_4 gain_5
## 187   T04     50      0     99     50     50
## 207   T06      0     50     99      0     50
## 227   T08     99      0     50     50     50
## 237   T09     99     99     50     50     50
##
## ** Bad players in R1
## In session_02 , player B3 has played:
##     round gain_1 gain_2 gain_3 gain_4 gain_5
## 165   T02      0     99      0     99     99
## 218   T07      0     99     99     99     50
##
## ** Bad players in R2
## In session_01 , player B2 has played:
##     round gain_1 gain_2 gain_3 gain_4 gain_5
## 127   T13     99     50     99     99     99
```

Remark: we did not apply any corrections for these individuals and kept them in the analysis.

2 Statistical analysis of Stigmer 15

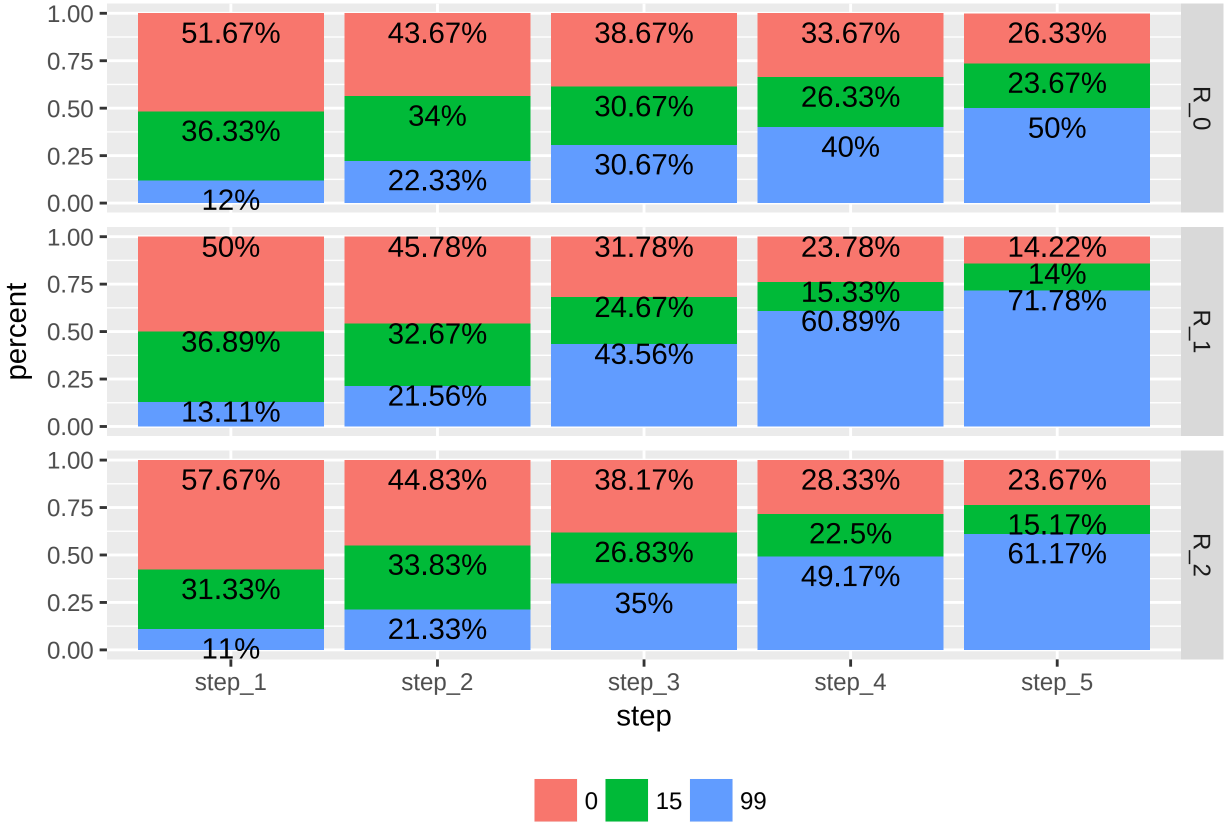
2.1 Statistical distribution of the gain per step

The idea here is to represent the empirical probability to get one of the values 0, 15 (or 50), 99 at each of the 5 steps, depending on the rule (0, 1 or 2).

2.1.1 Representation

We created the function *distib_per_step()* (codes presented in .Rmd file) which plots the statistical distribution depending on the Rule (R0, R1, R2) and the steps (1, 2, 3, 4 or 5).

```r
distib_per_step(fic = file_15, game = 15)
```



Remark: at step 1 and 2, the distributions seem to be the same under R0, R1 and R2. From step 3, we notice some differences between the rules. For example, at step 3 under R0, the probablity to find 99 or 15 is the same and slightly smaller than the probability to find 0. On the contrary, under R1, the probability to find 99 is higher than the probabilities to get 0 or 15. The distribution under R2 seem to be a mixture of the distributions under R0 and R1.

We will now present the analysis rule by rule.

2.2 Analysis under R0

We have selected the corresponding rows:

```r
file_15_R0 <- file_15[file_15$rule == "R0", ]
```

Under R0, there are :

- 2 sessions,
- There are 15 rounds in a session,
- For each session, there are two parallel games: 5 players called $A1, ..., A5$ and 5 players called $B1, ..., B5$, which means that in total there are 20 different players.

In this section, we try to identify what is the best strategy for players to optimize their profit. Then, we check if the players did adopt such a strategy and if we can see differences in terms of gain between the different strategies.

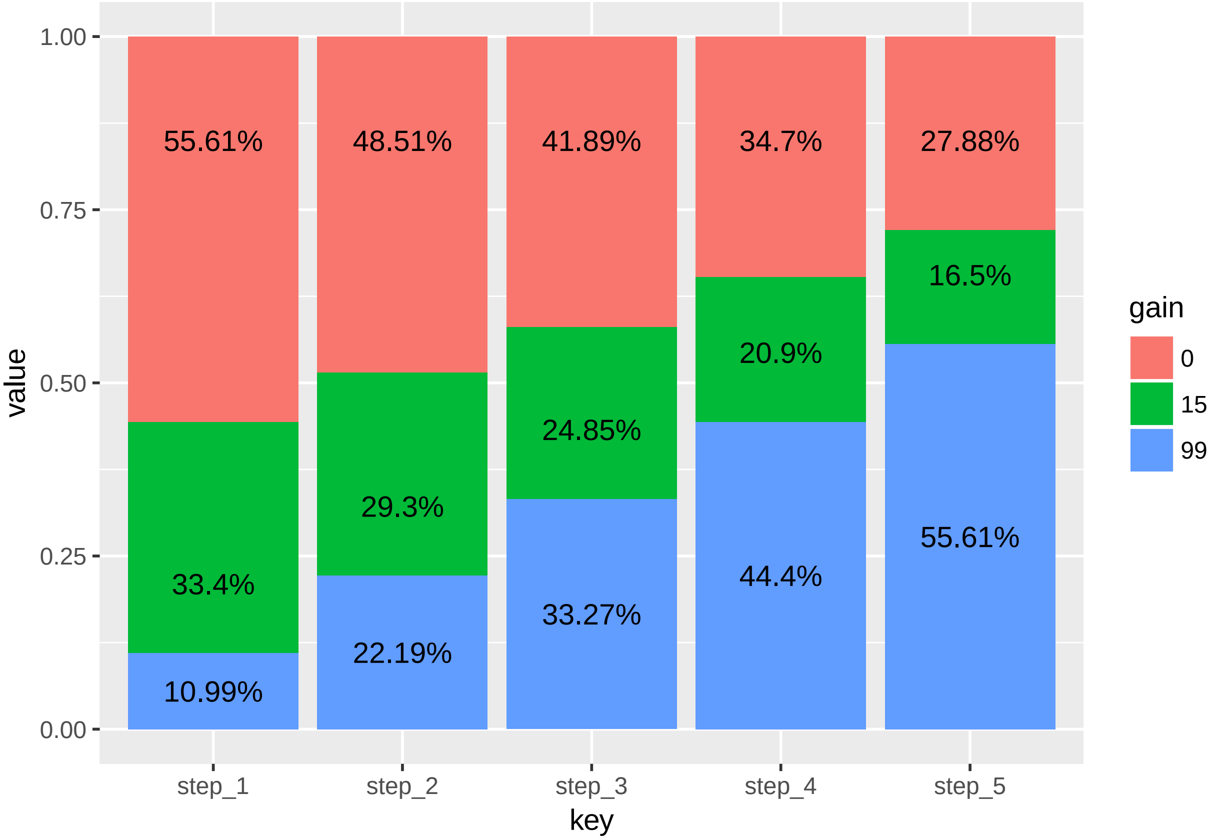
2.2.1 Probability to get 0, 15, 99 under R0: comparison between empirical and expected distribution

Under R0, players have no information about the rest of the group. Thus, to maximize their profit, a player should explore a new cell at each new step until they find the value 99 (this could be proved by using theoretical probability). Thus, we can simulate a high number of times this kind of behaviour, so that we obtain the expected distribution. We program the function *simu_distrib_R0()* (codes presented in .Rmd file) for doing this task.

We simulated the behaviours of 100,000 players (codes presented in .Rmd file).

We tidy the data (codes presented in .Rmd file).

We finally plot the theoretical distribution (codes presented in .Rmd file):



We will compare below the empirical distribution with the theoretical one obtained previously. We use a χ^2 test which consists of comparing at each step the empirical distribution of 0, 15, 99 to the theoretical one computed previously. It seems that the more we progress in the experience, the less the players seem to behave as players who optimize their profit. This is probably due to the fact that some players prefer to conserve their results by playing the value 15 (when they know where it is located) rather than continue to explore, especially at the end of a round.

Interpretation of the χ^2 test: the null hypothesis is "The distributions of empirical and theoretical are identical". When the p-value is lower than 0.05, we usually reject the null hypothesis. If the p-value is upper than 0.05, we cannot reject it. In our case, the distributions (empirical VS theoretical) are the same at step 1, 2, 3 and slightly different at step 4, 5.

```
chisq.test(table(as.factor(file_15_R0[, "gain_1"])),
           p = tab_15[, 1])
```

```
##
##  Chi-squared test for given probabilities
##
## data:  table(as.factor(file_15_R0[, "gain_1"]))
## X-squared = 1.8893, df = 2, p-value = 0.3888
```

```r
chisq.test(table(as.factor(file_15_R0[, "gain_2"])),
           p = tab_15[, 2])
```

```
##
##  Chi-squared test for given probabilities
##
## data:  table(as.factor(file_15_R0[, "gain_2"]))
## X-squared = 3.7163, df = 2, p-value = 0.156
```

```r
chisq.test(table(as.factor(file_15_R0[, "gain_3"])),
           p = tab_15[, 3])
```

```
##
##  Chi-squared test for given probabilities
##
## data:  table(as.factor(file_15_R0[, "gain_3"]))
## X-squared = 5.4413, df = 2, p-value = 0.06583
```

```r
chisq.test(table(as.factor(file_15_R0[, "gain_4"])),
           p = tab_15[, 4])
```

```
##
##  Chi-squared test for given probabilities
##
## data:  table(as.factor(file_15_R0[, "gain_4"]))
## X-squared = 5.6395, df = 2, p-value = 0.05962
```

```r
chisq.test(table(as.factor(file_15_R0[, "gain_5"])),
           p = tab_15[, 5])
```

```
##
##  Chi-squared test for given probabilities
##
## data:  table(as.factor(file_15_R0[, "gain_5"]))
## X-squared = 11.28, df = 2, p-value = 0.003553
```

2.2.2 Analysis of the gain per round under R0

We are now interested in the gain per round (the sum of the 5 gains obtained during a round).

2.2.2.1 Comparison between empirical and expected distributions

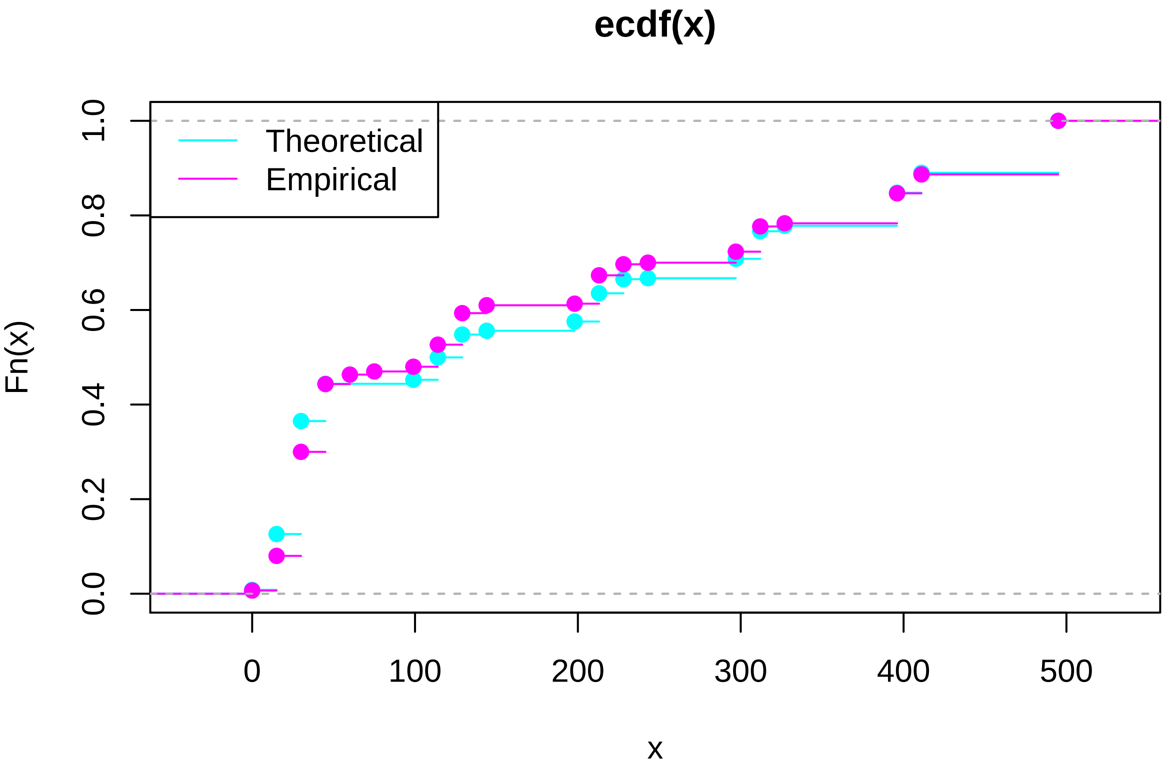
Thanks to the previous simulation, we can obtain the statistical distribution of the gain per round under R0 when players adopt an optimal strategy

Example of interpretation of the empirical cumulative distribution function :

- the probability to obtain a gain per round lower than or equal to 30 is 36.41%.
- 49.925% of the population obtained a value lower than or equal to 114. On the contrary, $100 - 49.925 =$ 50.075% obtained a gain per round strictly larger than 114.

```
cumsum(prop.table(table(final_gain_15)))
```

```
##       0      15      30      45      99     114     129     144     198
## 0.00802 0.12616 0.36513 0.44387 0.45240 0.49975 0.54807 0.55601 0.57563
##     213     228     243     297     312     327     396     411     495
## 0.63540 0.66511 0.66733 0.70820 0.76658 0.77809 0.84795 0.89012 1.00000
```



The plot of the empirical cumulative distribution function ("theoretical" versus "empirical") does not show a big difference between the two distributions since the two curves (cyan vs magenta) seem close.

We use the Kolmogorov-Smirnov test to verify that the two distributions (empirical and theoretical) are extracted from the same distribution or not. The null hypothesis which is "the two distributions are identical" cannot be rejected here. In other terms, the gain per round obtained by players could be the one obtained by players who optimize their profit.

```
ks.test(x = final_gain_15,
        y = file_15_R0[, "gain"],
        exact = F)
```

```
## Warning in ks.test(x = final_gain_15, y = file_15_R0[, "gain"], exact = F):
## p-value will be approximate in the presence of ties

##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  final_gain_15 and file_15_R0[, "gain"]
## D = 0.06513, p-value = 0.158
## alternative hypothesis: two-sided
```

Remark: note that the Kolmogorov-Smirnov test is supposed to be used on continous variable which is not the case here (the gain per round is indeed discrete). A solution could be to use instead a χ^2 test. However, we cannot use it for the following reason : in the optimal situation (theoretical distribution), players might not be able to obtain a gain per round equal to 60 or 75 (a player is exploring new cells unless they found 99, so they can only find the value 15 one, two or three times). Besides, the probability to obtain these values 60 or 75 are equal to 0, although these situations can occur in the experimental context (players who play several times the same cell containing 15). As the χ^2 test can be seen as the sum of the distances between theoretical and empirical values divided by the theoretical probability, we cannot use theoretical probability equal to 0.

2.2.2.2 Exploring or taking no risk ?

We show previously that players seem to behave as players who maximise their profit (exploring until the value 99 is found). However, we are interested in checking if all players behave like that and if not, could we notice differences in the final gain. For each player, we compute the final gain after the 15 rounds and compute the percentage of time they have chosen to continue to explore new cells until they find 99.

For doing that, we program the function *exploring()* (codes presented in .Rmd file) that will be also used in next sections. This functions permits to identify players who have the opportunity to explore at a given step and do it (or not).

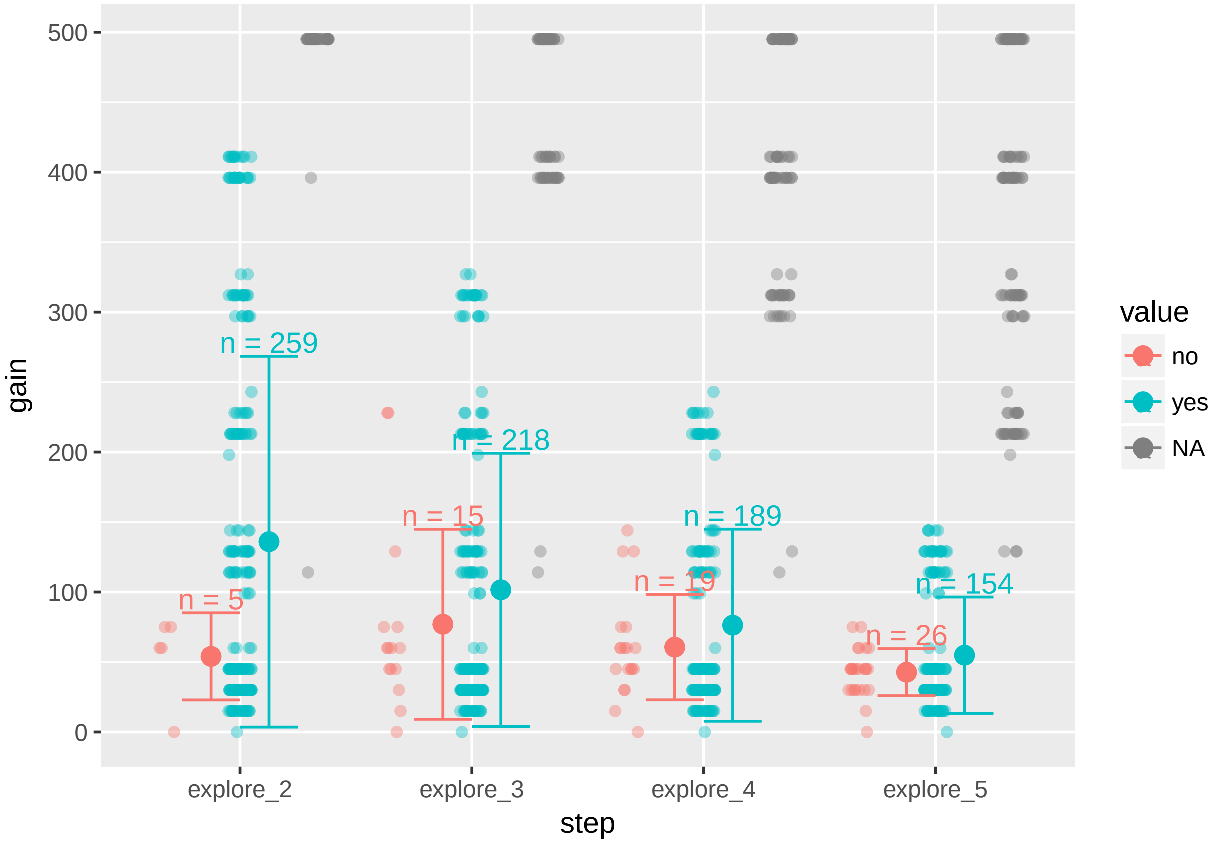
Remark: a player who does not explore is necessarily a player who plays a cell which has already been visited and does not contain the value 99. Typically, one could think about a player who prefers to play again the cell 15 rather exploring.

Now, we propose to compare the gain per round obtained depending on the fact that a player had the opportunity to explore new cells.

First, we tidy the data (codes presented in .Rmd file).

Then, we compute the mean and standard deviation obtained at each step and depending on the fact that a player explored new cells or not (codes presented in .Rmd file).

Finally, we plot in y-axis the gain per round and in x-axis the steps. We represent in blue (resp. in red) the gain per round obtained when a player explored new cells (resp. when he did not explore) knowing that the player had the opportunity to do it. We plot the gain per round in grey obtained by people who did not have the opportunity to explore (that means that they know where the cell 99 is located.



Interpretation of the plot:

- at each step there are much more people who prefer exploring ($n = 259, 218, 189, 154$ at step 2,3,4,5) rather taking no risk ($n = 5, 15, 19, 26$ at step 2,3,4,5).
- players who explore seem to obtain "in average" a higher gain per round than people who does not explore.
- the range of the standard error of the red part is always included in the range of the standard error of the blue part which seems to indicate that the differences between the means are not significant.

Statistical test:

To test the hypothesis of equality of means at each step, we use a non parametric test called "Kruskal-Wallis Rank Sum Test". It shows us that the differences of the means between players who explore and players who do not explore are non significant at any step.

```
(kruskal.test(gather_file_15$gain[gather_file_15$step == "explore_2"] ~
                                 factor(gather_file_15$value[gather_file_15$step == "explore_2"])))
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  gather_file_15$gain[gather_file_15$step == "explore_2"] by factor(gather_file_15$value[gather_
## Kruskal-Wallis chi-squared = 0.40938, df = 1, p-value = 0.5223
```

```
(kruskal.test(gather_file_15$gain[gather_file_15$step == "explore_3"] ~
                                 factor(gather_file_15$value[gather_file_15$step == "explore_3"])))
```

```
##
##  Kruskal-Wallis rank sum test
```

```
##
## data:  gather_file_15$gain[gather_file_15$step == "explore_3"] by factor(gather_file_15$value[gather_
## Kruskal-Wallis chi-squared = 0.049873, df = 1, p-value = 0.8233
```

```
(kruskal.test(gather_file_15$gain[gather_file_15$step == "explore_4"] ~
                             factor(gather_file_15$value[gather_file_15$step == "explore_4"])))
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  gather_file_15$gain[gather_file_15$step == "explore_4"] by factor(gather_file_15$value[gather_
## Kruskal-Wallis chi-squared = 0.54774, df = 1, p-value = 0.4592
```

```
(kruskal.test(gather_file_15$gain[gather_file_15$step == "explore_5"] ~
                             factor(gather_file_15$value[gather_file_15$step == "explore_5"])))
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  gather_file_15$gain[gather_file_15$step == "explore_5"] by factor(gather_file_15$value[gather_
## Kruskal-Wallis chi-squared = 0.27042, df = 1, p-value = 0.603
```

2.2.3 The effect of exploring (or not) on the final gain

In this section, we aggregate the results per player and consider thus the final gain. Indeed, we suspect that when a player adopts the same strategy during the 15 rounds, this probably has a effect on the final gain.

2.2.3.1 The expected "optimal" final gain

Thanks to the following theoretical distribution of the gain per round :

```
## final_gain_15
##       0      15      30      45      99     114     129     144     198
## 0.00802 0.11814 0.23897 0.07874 0.00853 0.04735 0.04832 0.00794 0.01962
##     213     228     243     297     312     327     396     411     495
## 0.05977 0.02971 0.00222 0.04087 0.05838 0.01151 0.06986 0.04217 0.10988
```

we can deducce what is the expected gain per round under R0 for an optimal behaviour. It corresponds simply to $\sum_x xPr[X = x]$ where x corresponds to the possible values of the gain per round. It is here equal to:

```
sum(as.numeric(names(prop_tab)) * prop_tab)
```

```
## [1] 183.5364
```

When mutiplying this number by 15, we obtain the theoretical optimal final gain

```
(opti_15_R0 <- 15 * sum(as.numeric(names(prop_tab)) * prop_tab))
```

```
## [1] 2753.046
```

We rank here the final gain obtained by the 20 players :

```
sort(players_15$gain)
```

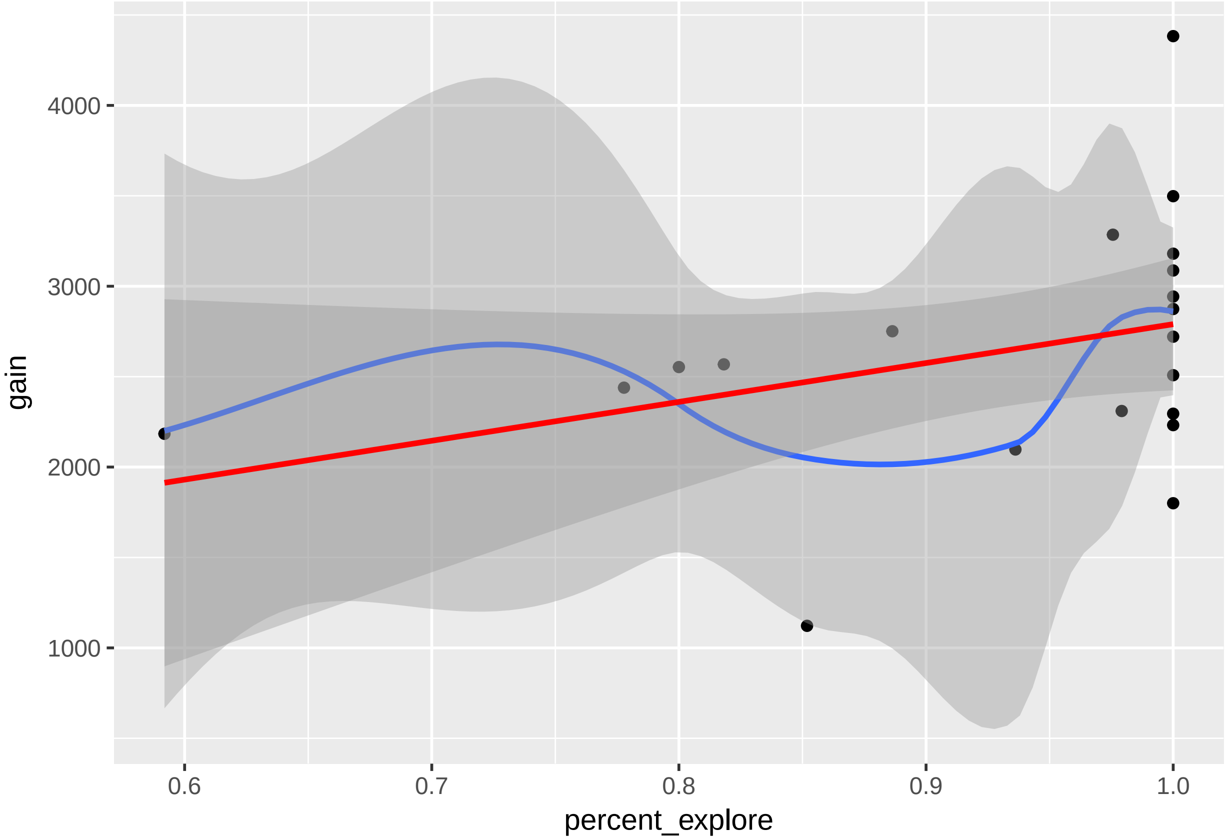
```
##  [1] 1122 1800 2097 2184 2232 2295 2310 2439 2508 2553 2568 2721 2751 2874
## [15] 2943 3087 3180 3285 3498 4383
```

We can remark that there are 7 players who obtained a final gain upper than the theoretical optimal final gain.

2.2.3.2 Linear regression

We plot the final gain depending on the percentage of times a player has explored new cells (knowing that he had the opportunity to do it).

We then plot the data :



We remark that :

- 11 players (on a total of 20) systematically explored when they had the opportunity to do it,
- the player with the highest score is a player who had explored new cells every time he had the opportunity to do it,
- the player with the lowest score is a player who had not explored new cells every time he has the possibility to do it
- the linear regression line seems to indicate that the more the player explore new cells the more his final gain will be higher. However, it seems difficult to give interestings results because globally, players tend to explore new cells when they can do it. Only few of them did not adopt such a strategy.

2.2.4 Conclusion under R0

Under R0, players had two main choices :

- exploring until they find 99 (the best strategy),
- takes no risk and plays again the value 15 when they already found it.

It seems that player globally tend to optimize their profit by exploring new cells until they find the value 99. Some of them have a tendency to keep the value 15 rather exploring. In theory, the first strategy is the best. However, because there are only a few observations in our experience, we could not detect significant differences between these two main behaviours at the different levels of observations (per round or per session).

2.3 Analysis under R1

We select the corresponding rows:

```
file_15_R1 <- file_15[file_15$rule == "R1", ]
```

Under R1, there are :

- 3 sessions,
- There are 15 rounds in a session,
- For each session, there are two parallel games: 5 players called $A1, ..., A5$ and 5 players called $B1, ..., B5$,which means that there are 30 different players.

In this section, we try to identify what is the best strategy for players. Under R1, players can use an additionnal information related to the frequency of visited cells. We try here to understand what is the best way to use this additional information and check if players do adopt this strategy.

2.3.1 Probability to get 0, 15, 99 under R1: comparison between empirical and expected distribution

2.3.1.1 Intuitive best strategy under R1

In this section we try to understand what could be the best strategy for players under R1.

2.3.1.1.1 At step 1

Players explore like under R0.

2.3.1.1.2 At step 2

We think about at least 2 different strategies :

- Strategy 1: a player (who did not find 99 yet) explores new cells like under R0 (he does not take into account what the other players have done).
- Strategy 2: a player (who did not find 99 yet) explores new cells among the cells which have not been visited at step 1 by the other players.

We do simulations to understand better what are the differences between these two strategies and awnser to several questions (see codes in the .Rmd file).

Q1: what is the probability that at least one player found the value 99 at the end of step 2?

- in strategy 1, this probability is equal to :

```
mean(res_sim_R1_s1[3, ])
```

```
## [1] 0.71453
```

- in strategy 2, this probability is equal to :

```
mean(res_sim_R1_s2[3, ])
```

```
## [1] 0.81452
```

Q2: what is the probability that the cell which has been the most visited is the cell with value 99?

- in strategy 1, this probability is equal to :

```r
sum(table(res_sim_R1_s1[1, ], res_sim_R1_s1[2, ])[,2])/100000
```

```
## [1] 0.28693
```

- in strategy 2, this probability is equal to :

```r
sum(table(res_sim_R1_s2[1, ], res_sim_R1_s2[2, ])[,2])/100000
```

```
## [1] 0.28912
```

Q3: what is the probability to find 99 when playing the k first cells the most visited ?

- interpretation with strategy 1 : the proba to find 99 by playing the most visited cell is equal to 28.7%. After playing the two most visited cell, this proba is equal to 51.7%

```
##       1       2       3       4       5       6       7       8       9
## 0.28693 0.51725 0.61129 0.70112 0.72275 0.78182 0.90695 0.98790 1.00000
```

- in strategy 2 : the proba to find 99 after playing the two most visited cell is higher than in strategy 1.

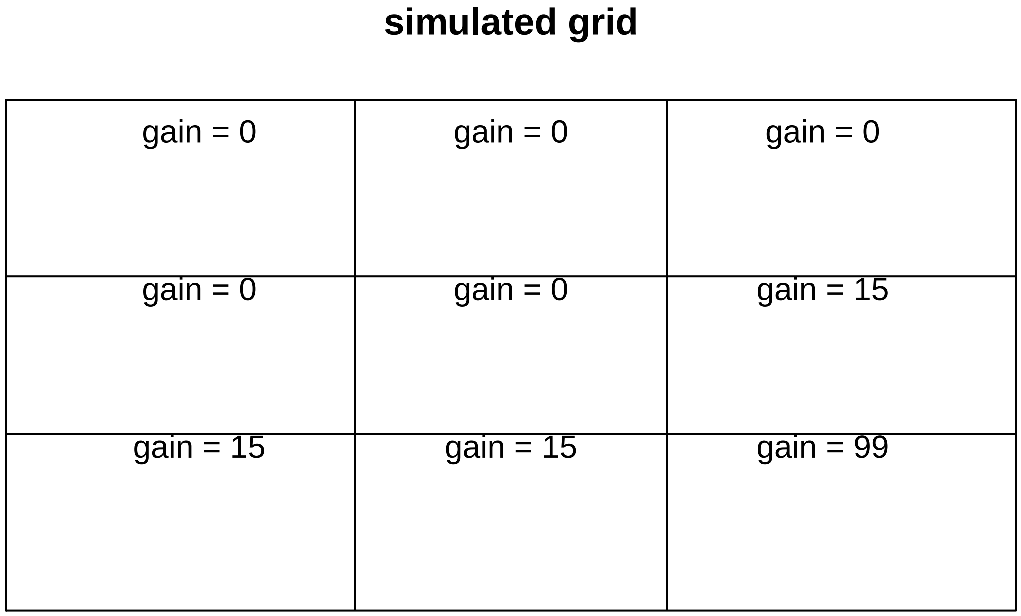
```
##       1       2       3       4       5       6       7       8       9
## 0.28912 0.63899 0.76593 0.81098 0.81485 0.82355 0.87076 0.95918 1.00000
```

Conclusion: strategy 2 seems more interesting than strategy 1. However, it could be than a mixture between both strategy (some players who adopt strategy 1 and some others who adopt strategy 2) would be the best thing to do. Let imagine a situation where after the 1st step, 5 different cells have been visited. That means that 4 cells must be visited such that the cell 99 will be necessarly found. However, it also means than the cell 99 has a higher probability (5/9) to be found among the 5 cells already visited. In that case, this is not completly obvious how should behave the players.

2.3.1.1.3 At step 3

A players chooses the cell which has been the most visited after the second step. Here we show why it seems the best solution to do.

For simplification, let consider a game with 5 players. The simulated grid is the following one :

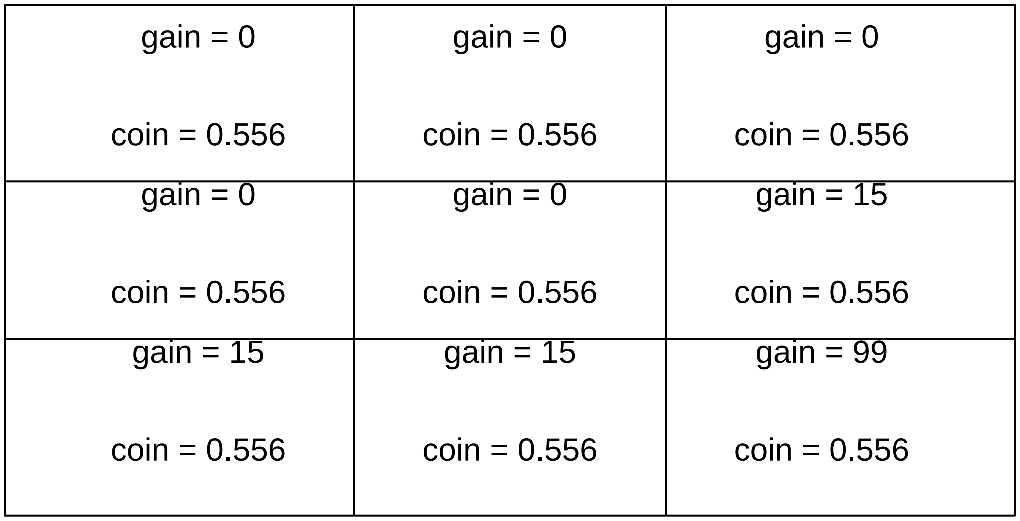


simulated grid

gain = 0	gain = 0	gain = 0
gain = 0	gain = 0	gain = 15
gain = 15	gain = 15	gain = 99

At the first tour, players are supposed to explore independantly the game. The expectation of the number of coins per cell is the following one (as there are 5 players, the sum of the expectations is equal to 5) :

Information given after step 1



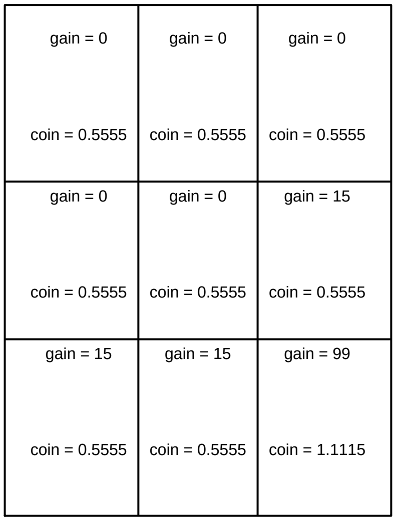
gain = 0
coin = 0.556
	gain = 0
	coin = 0.556
		gain = 0
		coin = 0.556

gain = 0
coin = 0.556
	gain = 0
	coin = 0.556
		gain = 15
		coin = 0.556

gain = 15
coin = 0.556
	gain = 15
	coin = 0.556
		gain = 99
		coin = 0.556

At the second step, players who found 99 are supposed to play 99 again. The others players are supposed to explore new cells with a probability equal to 1/8 (under strategy 1, for strategy 2 I am not sure it is identical) and the expected number of coins let in each cell is thus equal to $5 \times (1 - 1/9) \times 1/8$. After the second step, we sum the expected coins at the 1st and 2nd tour, such that we obtain the expected information which is given to all players before the 3rd step :

Proba at step 2

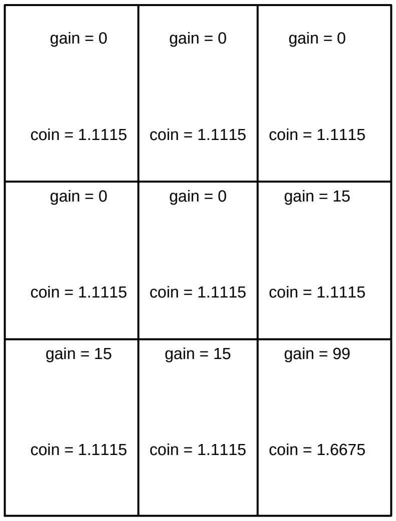


gain = 0
coin = 0.5555
	gain = 0
	coin = 0.5555
		gain = 0
		coin = 0.5555

gain = 0
coin = 0.5555
	gain = 0
	coin = 0.5555
		gain = 15
		coin = 0.5555

gain = 15
coin = 0.5555
	gain = 15
	coin = 0.5555
		gain = 99
		coin = 1.1115

Information given after step 2



gain = 0
coin = 1.1115
	gain = 0
	coin = 1.1115
		gain = 0
		coin = 1.1115

gain = 0
coin = 1.1115
	gain = 0
	coin = 1.1115
		gain = 15
		coin = 1.1115

gain = 15
coin = 1.1115
	gain = 15
	coin = 1.1115
		gain = 99
		coin = 1.6675

We notice that the cell 99 might be the cell with the maximum expected number of coins let at the end of step 2. This suggests that to optimize their profit in R1, players should play the cell which has been the most visited after the second step.

Moreover, the simulation results obtained at step 2 tend to indicate that if a player has already visited the cell the most visited and knows that it does not contain 99, he should try to play the second most visited cell.

Finally, if the player knows than the second most visited is not the good one (he has already played the two most visited cells and knows that they do not contain 99), than he should visit new cell which has not been yet explored by the group.

We now simulate such a behaviour to understand which could be the best strategy to adopt after step 3.

Q1: what is the probability that at least one player found the value 99 at the end of step 3?

```
mean(res_sim_R1_3step[3, ])
```

```
## [1] 0.88108
```

Q2: what is the probability that the cell which has been the most visited is the cell with value 99?

```
sum(table(res_sim_R1_3step[1, ], res_sim_R1_3step[2, ])[,2])/100000
```

```
## [1] 0.51264
```

Q3: what is the probability to find 99 when playing the k first cells the most visited ?

```
##       1       2       3       4       5       6       7       8       9
## 0.51264 0.60307 0.86001 0.87900 0.88108 0.88228 0.89913 0.95252 1.00000
```

Remark: when a player has already played the most visited cell and knows that it does not contain the value 99, it could be more interesting for him to play the 3rd most visited cell rather the 2nd one.

2.3.1.1.4 At step 4

A player visits the most visited cell if he did explore it yet. Otherwise, he tries to visit the second most visited cell. Otherwise, he tries the 3rd most visited cell. Otherwise he visits a cell which has not been explored. If they have all been explored, he chooses one of them randomly.

We now simulate such a behaviour to understand which could be the best strategy to adopt after step 3.

Q1: what is the probability that at least one player found the value 99 at the end of step 4?

```
mean(res_sim_R1_4step[3, ])
```

```
## [1] 0.95259
```

Q2: what is the probability that the cell which has been the most visited is the cell with value 99?

```
sum(table(res_sim_R1_4step[1, ], res_sim_R1_4step[2, ])[,2])/100000
```

```
## [1] 0.77964
```

Q3: what is the probability to find 99 when playing the k first cells the most visited ?

```
##       1       2       3       4       5       6       7       8       9
## 0.77964 0.79304 0.81440 0.94778 0.95256 0.95259 0.95294 0.96046 1.00000
```

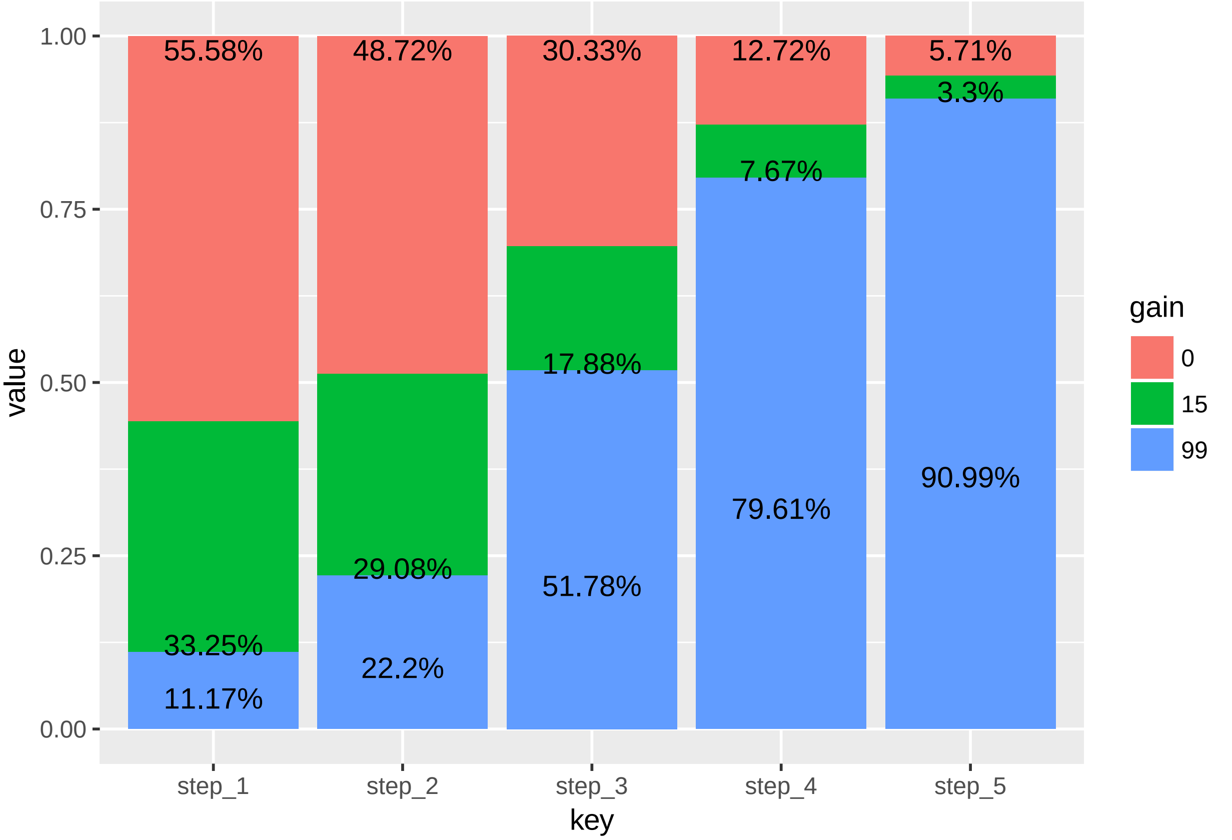
2.3.1.1.5 At step 5

Like in step 4. If players has already played the four first most visited cell and he knows that they do not contain 99, he explores a cell which has not been explored. If they have all been explored, he chooses one of them randomly. We simulate such a behaviour in the next section.

2.3.1.2 Simulation of the optimal behaviour of players under R1

We now simulate a full party as described previously. This function is called *simu_distrib_R1()* (codes presented in .Rmd file).

We obtain this distribution. Note that at step 1 and 2, distributions are obviously identical to the ones obtained in Rule 0.



We compare below the empirical distribution with the theoretical one obtained previously. We use a χ^2 test which consists in comparing at each step the empirical distribution of 0, 15, 99 to the theoretical one computed previously. At step 2, players behave as we can expect. At step 3, 4 and 5, the p-values of the test are lower than 5% which indicates that players do not behave as players who optimize their profit.

```
chisq.test(table(as.factor(file_15_R1[, "gain_2"])),
           p = tab_15_R1[, 2])
```

```
##
##  Chi-squared test for given probabilities
##
## data:  table(as.factor(file_15_R1[, "gain_2"]))
## X-squared = 2.8747, df = 2, p-value = 0.2376
```

```
chisq.test(table(as.factor(file_15_R1[, "gain_3"])),
           p = tab_15_R1[, 3])
```

```
##
##  Chi-squared test for given probabilities
##
## data:  table(as.factor(file_15_R1[, "gain_3"]))
## X-squared = 17.766, df = 2, p-value = 0.0001387
```

```
chisq.test(table(as.factor(file_15_R1[, "gain_4"])),
           p = tab_15_R1[, 4])
```

```
##
```

```
##  Chi-squared test for given probabilities
##
## data:  table(as.factor(file_15_R1[, "gain_4"]))
## X-squared = 97.46, df = 2, p-value < 2.2e-16
```

```
chisq.test(table(as.factor(file_15_R1[,"gain_5"])),
           p = tab_15_R1[, 5])
```

```
##
##  Chi-squared test for given probabilities
##
## data:  table(as.factor(file_15_R1[, "gain_5"]))
## X-squared = 231.17, df = 2, p-value < 2.2e-16
```

2.3.2 Analysis of the gain per round under R1

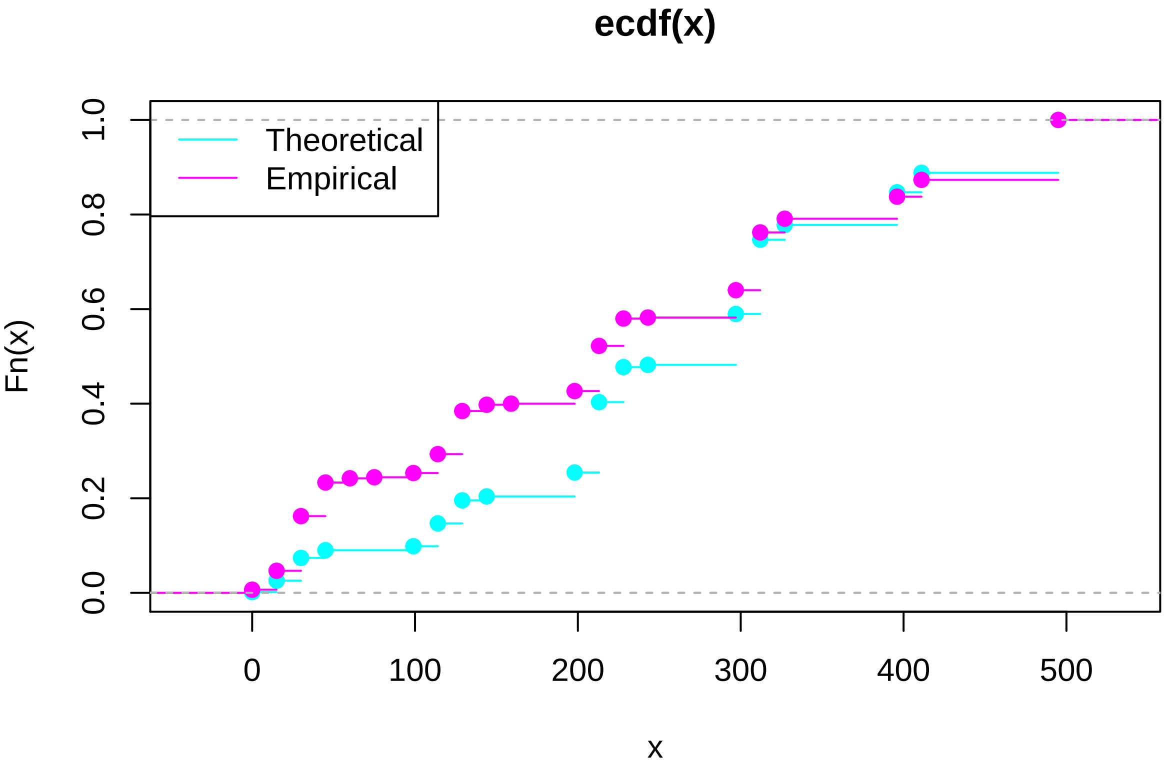
We are now interested in the gain per round (the sum of the 5 gains obtained during a round).

2.3.2.1 comparison between empirical and expected distribution

Thanks to the previous simulation, we can obtain the statistical distibution of the final gain under R1 when players adopt an optimal behaviour.

The plot of the empirical cumulative distribution function ("theoretical" versus "empirical") does not show a big difference between the two distributions.

```
##       0       15      30      45      99     114     129     144     198
## 0.00161 0.02585 0.07390 0.09014 0.09859 0.14695 0.19558 0.20394 0.25457
##     213     228     243     297     312     327     396     411     495
## 0.40329 0.47727 0.48218 0.58965 0.74675 0.77801 0.84713 0.88830 1.00000
```



The following test confirms the previous plot. Hereafter, the test rejects the hyptohesis of equality of the two distributions.

```
ks.test(x = final_gain_15_R1,
        y = file_15_R1[, "gain"])
```

```
## Warning in ks.test(x = final_gain_15_R1, y = file_15_R1[, "gain"]): p-value
## will be approximate in the presence of ties

##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  final_gain_15_R1 and file_15_R1[, "gain"]
## D = 0.19606, p-value = 2.22e-15
## alternative hypothesis: two-sided
```

2.3.2.2 Copying, exploring or taking no risk ?

The idea of this section is to determine whose players belong to one of these strategy (copying, exploring weakly, exploring strongly, taking no risk) and if we can notice differences in terms of gain between them.

Algorithm (step 1): to detect players who copy other players at each step, we need to know which is the most visited cell at the end of a step. We create a matrix which contains for each of the 9 cells, the number of times it has been visited at the end of a step for a given round. Then, we count the number of visits per cell at each step, for given players (A or B) per round and per session (codes presented in .Rmd file).

After, we create the function *cell_visited()* (codes presented in .Rmd file) which could be useful later. It consists in giving for each step the ranking of the most visited cells in a sense of the players have let a coin.

We apply the previous function to our data (codes presented in the .Rmd file).

Algorithm (step 2): once we kwow how many times the cells have been visited at each step, one could say if a player decides "yes" or "not" to play the most visited cell. We do the following assumption: if player has already played the most visited cell and knows that it does not contain the value 99, he is supposed to play the second most visited cell (from step 3 to step 5), then the third (from step 4 to step 5) and the fourth (at step 5). For doing this, we create the function *copy_or_not()* (codes presented in .Rmd file) which permits to know at step 2, 3, 4 and 5 if a player belongs to one of these categories:

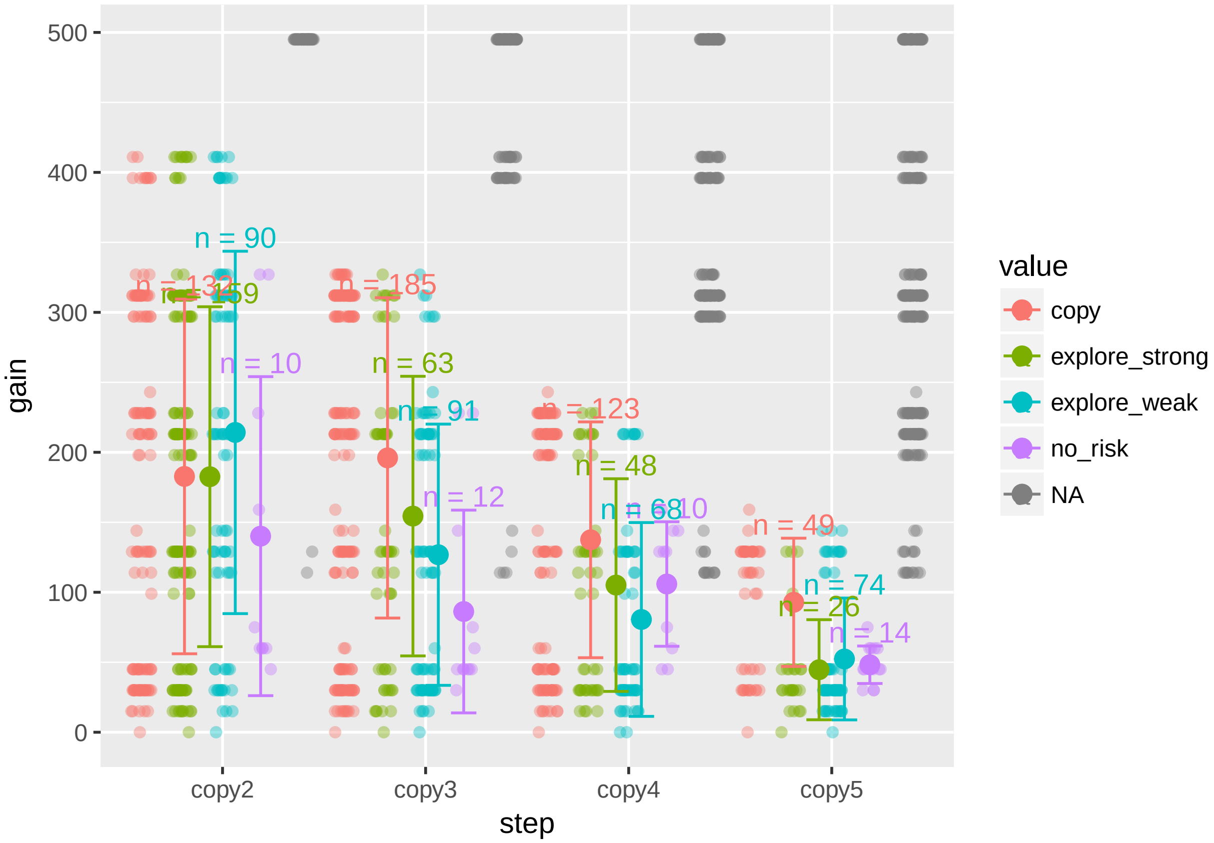
- **copy**: a player visits the cell the most visited (the cell which contains the maximum number of coins),
- **explore strongly**: a player played a cell which has never been visited,
- **explore weakly**: a player played a cell which has been visited by other players but is not among the most visited cells,
- **no_risk**: a player does not follow one of the three previous behaviours,

We apply our data to the previous function (codes presented in .Rmd file).

For representing the data, we first tidy the data (codes presented in .Rmd file).

We compute the mean and standard deviation obtained at each step and depending on the fact that a player explored new cells or not (codes presented in .Rmd file).

We plot in y-axis the gain per round and in x-axis the steps. We represent in red (resp. in green resp. in blue) the gain per round obtained when the player played the most visited cell visited (resp. when he explored resp. when he took no risk) knowing that the player had the opportunity to do it. We plot the gain per round in grey obtained by people who did not have the opportunity to visit (that means that they knew where the cell 99 was located).



Interesting remarks concerning the figure:

- at step 2, it is interesting to remark that players adopt very different behaviours. Moreover, many of them ($n = 132$) decided to play the most visited cell although there is obviously no reason to do it at this step . There are more people who prefer exploring strongly ($n = 159$) rather exploring weakly ($n = 90$). We can see that at this step, it seems more interesting to explore weakly. This is a result inverse to what we can expect in an optimal behaviour. That could be explained by the fact that people do not adopt a similar strategy all together.
- at step 3, 4 and 5 it is always more interesting to copy. Moreover, this is the choice which is the most followed by players (excepted at step 5).
- at step 5, there are more players who prefer exploring ($n = 97$) rather copying ($n = 52$). We canno't observe big differences between “exploring” and “taking no risk”.
- The worst case which can happen is a situation where people by copying, do not have the opportunity to explore enough and hence do not sucess to find the value 99. We present here a example where this situation occured :

```
file_15_R1_copy[file_15_R1_copy$round_p == "T05_A_session_01",
                c(paste0("gain_", 1:5), paste0("cell_", 1:5), paste0("copy", 2:5))]
```

```
##     gain_1 gain_2 gain_3 gain_4 gain_5 cell_1 cell_2 cell_3 cell_4 cell_5
## 341     15     15      0      0      0    1_1    2_0    1_0    2_2    0_2
## 342      0     15     15      0      0    0_0    1_1    2_0    1_0    2_2
## 343      0     15      0     15      0    2_2    1_1    1_0    2_0    0_1
## 344      0     15      0      0     15    1_0    1_1    0_1    2_2    2_0
## 345     15     15     15      0     15    1_1    1_1    1_1    1_0    1_1
##              copy2       copy3      copy4          copy5
## 341 explore_strong   explore_weak       copy explore_strong
## 342           copy   explore_weak         copy           copy
## 343           copy   explore_weak         copy   explore_weak
## 344           copy explore_strong explore_weak           copy
## 345        no_risk        no_risk         copy        no_risk
```

To test the hypothesis of equality of the means (here we test simultaneously the equality "copy" = "explore strongly" = "explore weakly" = "no_risk") at each step, we use a non parametric test called "Kruskal-Wallis Rank Sum Test". It shows us that the differences of the means are significant at step 3, 4 and 5. It is not at step 2.

```
(kruskal.test(gather_file_15_R1$gain[gather_file_15_R1$step == "copy2"] ~
                            factor(gather_file_15_R1$value[gather_file_15_R1$step == "copy2"])))
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  gather_file_15_R1$gain[gather_file_15_R1$step == "copy2"] by factor(gather_file_15_R1$value[ga
## Kruskal-Wallis chi-squared = 4.767, df = 3, p-value = 0.1897
```

```
(kruskal.test(gather_file_15_R1$gain[gather_file_15_R1$step == "copy3"] ~
                            factor(gather_file_15_R1$value[gather_file_15_R1$step == "copy3"])))
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  gather_file_15_R1$gain[gather_file_15_R1$step == "copy3"] by factor(gather_file_15_R1$value[ga
## Kruskal-Wallis chi-squared = 30.293, df = 3, p-value = 1.198e-06
```

```
(kruskal.test(gather_file_15_R1$gain[gather_file_15_R1$step == "copy4"] ~
                            factor(gather_file_15_R1$value[gather_file_15_R1$step == "copy4"])))
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  gather_file_15_R1$gain[gather_file_15_R1$step == "copy4"] by factor(gather_file_15_R1$value[ga
## Kruskal-Wallis chi-squared = 23.571, df = 3, p-value = 3.07e-05
```

```
(kruskal.test(gather_file_15_R1$gain[gather_file_15_R1$step == "copy5"] ~
                            factor(gather_file_15_R1$value[gather_file_15_R1$step == "copy5"])))
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  gather_file_15_R1$gain[gather_file_15_R1$step == "copy5"] by factor(gather_file_15_R1$value[ga
## Kruskal-Wallis chi-squared = 25.213, df = 3, p-value = 1.394e-05
```

2.3.3 The effect of copying VS exploring strongly VS exploring weakly VS re-playing 15 on the final gain

In this section, we aggregate the results per player and consider thus the final gain. There are 30 players and for doing a statistical analysis on these players, we need to define some variables for each player.

2.3.3.1 The expected "optimal" final gain

Thanks to the following theoretical distribution of the gain per round :

```
## final_gain_15_R1
##       0      15      30      45      99     114     129     144     198
## 0.00161 0.02424 0.04805 0.01624 0.00845 0.04836 0.04863 0.00836 0.05063
##     213     228     243     297     312     327     396     411     495
## 0.14872 0.07398 0.00491 0.10747 0.15710 0.03126 0.06912 0.04117 0.11170
```

we can deducce what is the expected gain per round under R1 for an optimal behaviour. It corresponds simply to $\sum_x xPr[X = x]$ where x corresponds to the possible values of the gain per round. It is here equal to:

```
sum(as.numeric(names(prop_tab)) * prop_tab)
```

```
## [1] 266.865
```

When mutiplying this number by 15, we obtain the theoretical optimal final gain

```
(opti_15_R1 <- 15 * sum(as.numeric(names(prop_tab)) * prop_tab))
```

```
## [1] 4002.975
```

We rank here the final gain obtained by the 20 players :

```
sort(players_15_R1$final_gain)
```

```
##  [1]  954 1761 2295 2745 2844 2928 2934 3033 3108 3171 3201 3225 3354 3384
## [15] 3528 3537 3567 3576 3720 3735 3744 3963 4071 4086 4092 4140 4194 4323
## [29] 4443 4635
```

We can remark that there are 8 players who obtained a final gain upper than the theoretical optimal final gain.

2.3.3.2 Create new variables per player

We create the following variables per player :

- The dependent variable is the final gain.
- The characterestics we are interested in are the behaviours of the players at step 2, 3, 4 and 5. For example, if we look at the player "A1" in session 01, we observe the following behaviour at step 2: he copied only 2 times whereas he explored "strongly" 8 times. Hence, this player will be considered as an "explorer_strongly" at step 2.

```
## 
##           copy explore_strong   explore_weak 
##              2              8              3
```

At step 3, he will be considered as a copyer (4 times "copy") :

```
## 
##           copy explore_strong   explore_weak 
##              4              2              2
```

Important remark: we can suppose that if some players change their behaviour, this is probably due to the fact that adapt their strategy to the present round. For example, depending on the maximum number of coins let on a cell, the player might play differently from a round to another. Such a behaviour is not taken into account in this part. Indeed, we create a variable based on a "general" behaviour at every step.

At step 4, he will be considered as a "copyer" :

```
## 
##           copy explore_strong 
##              4              2
```

At step 5, he will be considered as an “explorer_strong” :

```
table(file_15_R1_copy[file_15_R1_copy$session == "session_01" &
                  file_15_R1_copy$player == "A1", "copy5"])
```

```
##
## explore_strong   explore_weak
##              3              2
```

We do this for all players (codes presented in .Rmd file).

We know that there are 2 players who behave badly (see first section). We delete them for the statistical analysis because they have a too strong influence. Note they probably had an influence during the game.

```
players_15_R1 <- filter(players_15_R1, ! (player == "A3" &  session == "session_01") )
players_15_R1 <- filter(players_15_R1, ! (player == "B4" &  session == "session_03") )
```

2.3.3.3 Exploratory analysis

At step 2 : we analyse now the behaviours of the players at step 2. Most of them prefer exploring strongly (13 players) rather copying (12 players). Only one player prefer to keep the value 15 (note that this behaviour does not help the group because other players could then think that the cell 99 is at the cell the player is playing again).

```
table(players_15_R1$step_2)
```

```
##
##           copy explore_strong   explore_weak        no_risk
##             12             13              2              1
```

Besides, the average mean of the final gain obtained by the players who have a tendency to explore strongly (3644) is larger than the average mean of the group who copies (3528), explores weakly (3376) or takes no risk (3108).

The final gain per group :

```
tapply(players_15_R1$final_gain, players_15_R1$step_2, mean)
```

```
##           copy explore_strong   explore_weak        no_risk
##       3528.500       3644.077       3376.500       3108.000
```

At step 3 and 4:

```
table(players_15_R1$step_3, players_15_R1$step_4)
```

```
##
##                  copy explore_strong explore_weak no_risk
##   copy             18              2            3       1
##   explore_strong    1              0            0       0
##   explore_weak      1              0            2       0
```

we cross the behaviours at step 3 and 4 and create a new variable which consists in “copying” if players copy both at step 3 and 4, “others” if they do not only copy (codes presented in the .Rmd file).

There is a majority of players who copy (18) than do something different (10).

```
table(players_15_R1$step_3_and_4)
```

```
##
##   copy others
##     18     10
```

The average mean of the final gain is higher for players who do copy (3597) rather doing something else (3482) :

```
tapply(players_15_R1$final_gain, players_15_R1$step_3_and_4, mean)
```

```
##     copy   others 
## 3597.333 3482.400
```

At step 5: most of players prefer to explore weakly. The frequency :

```
table(players_15_R1$step_5)
```

```
## 
##           copy explore_strong   explore_weak        no_risk 
##              6              4             15              3
```

The final gain per group is higher for people who explore strongly (4008) rather taking no risk (3528), copying (3452) or explore weakly (3483).

```
tapply(players_15_R1$final_gain, players_15_R1$step_5, mean)
```

```
##           copy explore_strong   explore_weak        no_risk 
##         3452.0         4008.0         3483.2         3528.0
```

Behaviours adopted across the time: here we look if there is a better strategy to adopt during the round. For example, is it better to “explore stongly” at step 2, then “copy” at step 3/4 and finally “explore weakly” at step 5.

We present all the observed behaviours at step 2/step 3,4/step 5 and rank them with the average final gain:

```
##                              behaviour final_gain freq
## 1                       copy/others/copy     2745.0    1
## 2                   no_risk/copy/no_risk     3108.0    1
## 3    explore_weak/others/explore_weak     3225.0    1
## 4              explore_strong/copy/copy     3354.0    1
## 5                      copy/copy/no_risk     3384.0    1
## 6    explore_strong/copy/explore_weak     3411.0    3
## 7                 copy/copy/explore_weak     3411.6    5
## 8      explore_weak/copy/explore_weak     3528.0    1
## 9            explore_strong/others/copy     3550.0    3
## 10 explore_strong/others/explore_weak     3611.0    3
## 11              copy/others/explore_weak     3685.5    2
## 12              copy/copy/explore_strong     3910.5    2
## 13                        copy/copy/copy     3963.0    1
## 14          explore_strong/copy/no_risk     4092.0    1
## 15 explore_strong/copy/explore_strong     4105.5    2
```

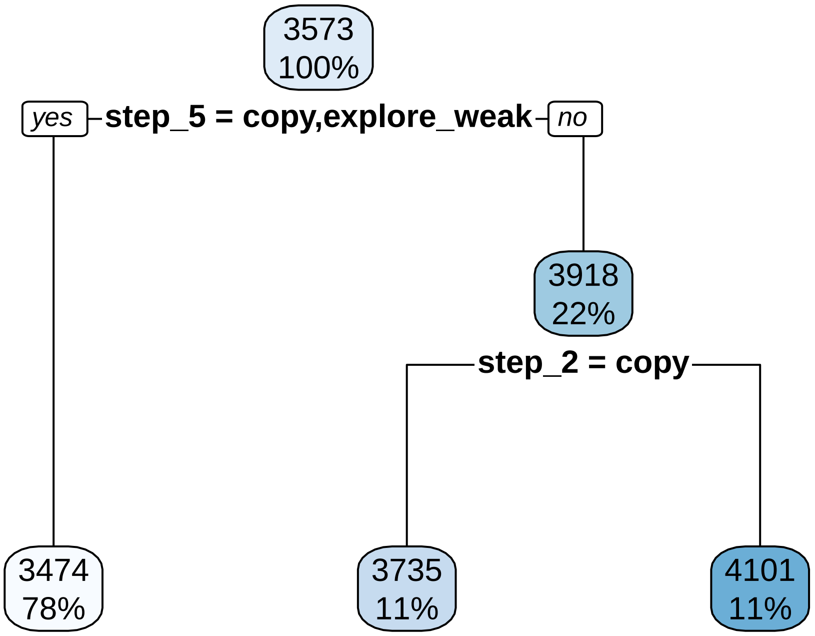
Interpretation: by considering our few number of observations, the remarks we are giving here might be used with precautions.

- the three worst strategy consists in copy/others/copy, no_risk/copy/no_risk, explore_weak/others/explore_weak.
- the two best strategy consists in explore_strong/copy/explore_strong and explore_strong/explore/explore_weak.

2.3.3.4 Machine Learning

Finally, we do a regression tree trying to explain the final gain obtained by players depending on their behaviours. Before doing this, we drop the player who took no risk at step 2 and 5 which seems to adopt a behaviour very different from the other players.

```
players_15_R1 <- filter(players_15_R1, step_2 != "no_risk")
```



Interpretation: at the first step, the tree discriminates the population into two groups, one group who does copy/explore weakly at step 5 (on the left) and another group who does not (it means that there are the players who explore strongly/takes no risk). The group on the right has a lower average mean (3918) than the group on the left (3474) which is a final node. The discrimination continues sub-group by sub-group until the final node which gives the percentage of observations and the average mean of the final gain. Usually, we look the observations which fall in the finale nodes with the highest (resp. lowest) predicted values. In our case, there three final nodes :

- Last node in dark blue whith the highest value (4101) : 3 players who explore strongly/takes no risk at step 5 and do not copy at step 2,
- Middle node with middle value : 3 players who explore strongly/takes no risk at step 5 and do copy at step 2,
- 1st node with small values (3474) : 21 players who copy or explore weakly at step 2.

2.3.4 Conclusion under R1

We propose an optimal behaviour under R1. It seems to give good results although it could be probably improved by taking into account some other factors such as the number of cell visited.

The main key under R1 seems to be able to explore the maximum number of cells at step 1 and 2 by visiting the cells which have not been visited yet and then to adopt a strategy of copying at step 3, 4, 5 when it is possible.

When copying at step 3, 4 and 5, the problem is that if the cell 99 has not been found, players will just visit the wrong cells. Thus, at step 5, it is possible that the cell 99 has not been visited yet. Hence, players should not hesitate to explore cells which have not been visited yet when they have already copied at previous steps without succeeding.

2.4 Analysis under R2

We select the corresponding rows:

```
file_15_R2 <- file_15[file_15$rule == "R2", ]
```

Under R2, there are :

- 4 sessions,
- For each session, there are two parallel games: 5 players called $A1, ..., A5$ and 5 players called $B1, ..., B5$,
- There are 15 rounds in a session.

At each step, the players have the opportunity to let (or not to let) a coin on the cell they have just played.

2.4.1 Optimal strategy for the group

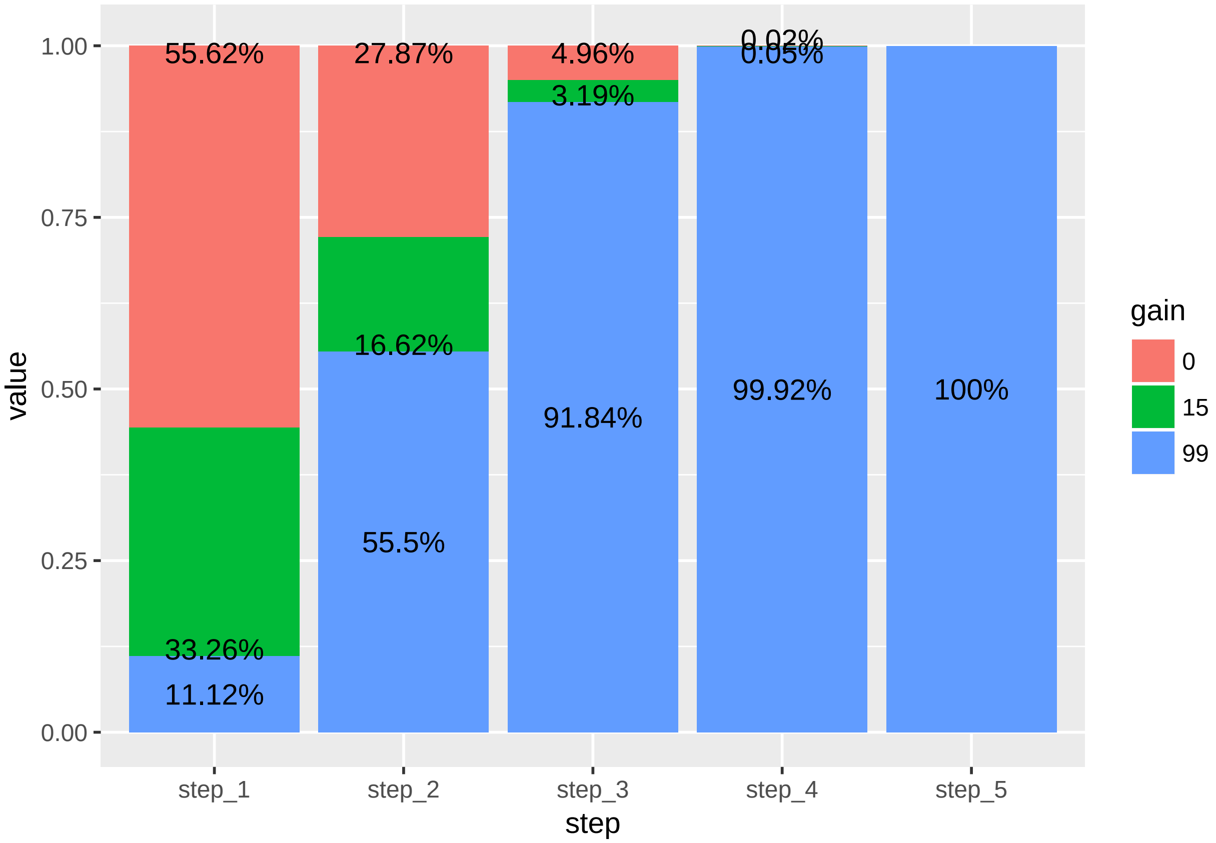
It seems that the best strategy for the group under R2 consists of :

- leaving a coin only and only if players found the cell 99.
- exploring cells which have not been visited until the cell 99 has been discovered by somebody of the group.

Let simulate such a behaviour by using the function *simu_distrib_R2()* (codes are in the .Rmd file).

We will replicate 100,000 sessions:

```
## Warning in gather_tab_15_R2$pos[gather_tab_15_R2$gain == "15"] <- 0.01
## + : le nombre d'objets à remplacer n'est pas multiple de la taille du
## remplacement
```



Remark: at step 5, the probability that a player found 99 is 1. Thus, this strategy is obviously advantageous for the group. Comparing to the empirical distribution, this is clearly not the behaviour which has been adopted by the players. This can be explained by the fact that if all players behave similarly, they optimize the total gain, but the probability to have the best score at the end of a session among the players is the same for all. Hence, if players first think about how they could obtain a better score than other, they might adopt a strategy which is different from the one adopted by others.

2.4.2 Example of an optimal strategy for one player

To illustrate our idea, we will simulate a game where 4 players adopt the previous strategy and one player decide to never leave a coin. However, he will adopt the strategy which consists of playing the most visited cell. In this situation, it is interesting to notice that other players will not necessarily remark than player 5 is rigging the game because he does not give a bad information. Here, we are interested to know the final gain this players will obtain at the end of the 15th round and what is his probability to win the session. We program the function *simu_gain_R2_b()* (codes availabes in the .Rmd file).

We will replicate 1000 times:

The probability for player 5 to win the session is much more higher for him/her than the other players :

```
##
##     1     2     3     4     5
## 0.122 0.127 0.139 0.119 0.493
```

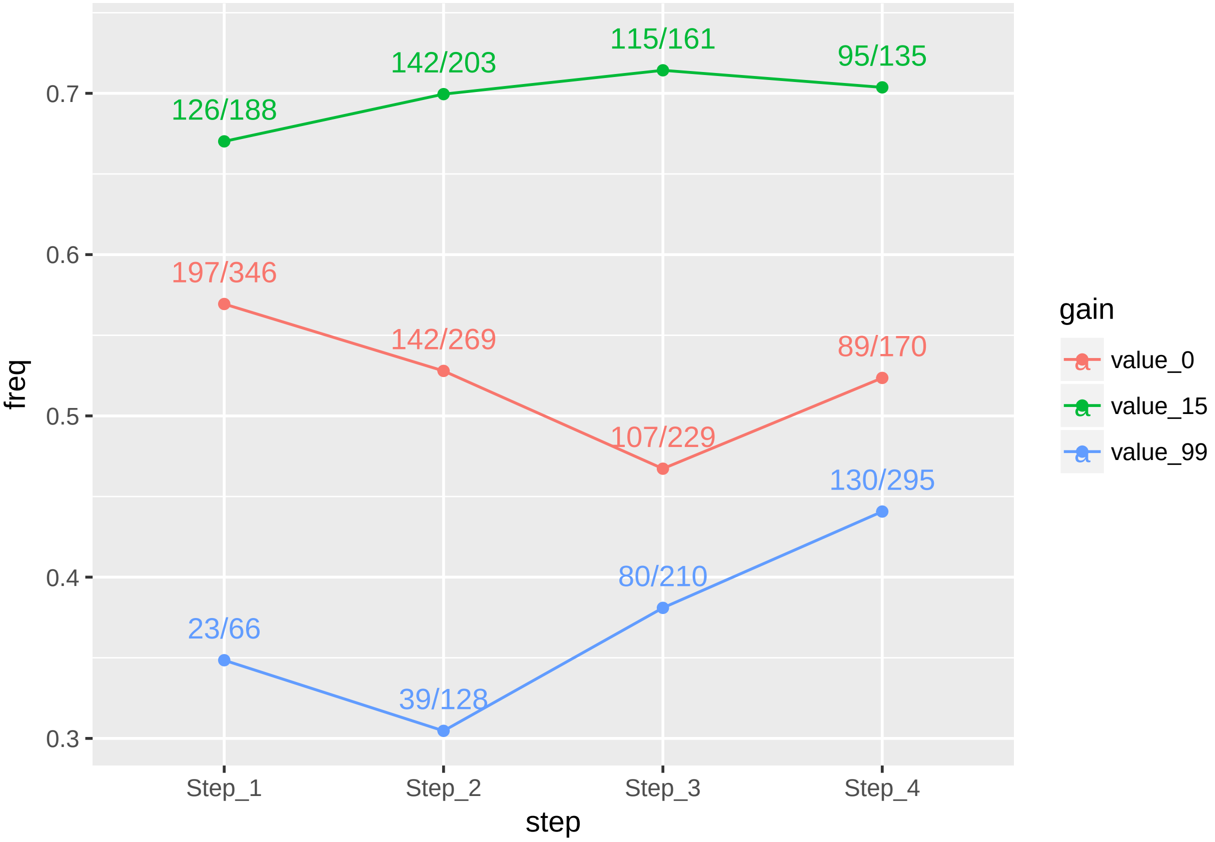
The average mean of the final gain is also quite higher for the player 5:

```
## [1] 5000.238 5005.698 4998.519 4993.674 5248.794
```

2.4.3 Analysis of the behaviours of the players under R2

We compute here the probability to leave a coin after playing the cell 99 depending on the step.

We plot the figure:



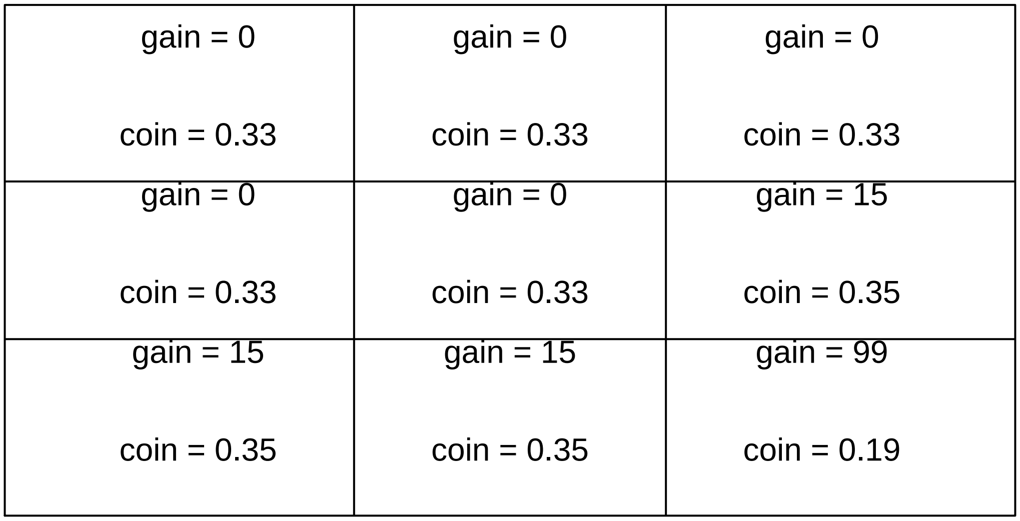
Interpretation: at step 1, 66 players found the value 99. 34.85% of them left a coin on the cell. Comparing to the value 0 and 15, it seems that players left more frequently a coin when they found 15 or 0 rather than 99. Thus, in these conditions, it is difficult to say when it is more interesting to play the most visited cell or to explore. This is what we are going to check now.

2.4.4 Analysis of the number of coins left depending on the values 0, 15 or 99.

From the previous figure, we can deduct the "empirical" expected number of coins left by players after each step. How did we do that : there are 197 players who let a coin after obtaining the value 0. To get the "empirical" expected number of coins for any cell which contains 0, we simply divide 197 by 120 ($120 = 15$ rounds $\times$ 2 parallel session $\times$ 4 different sessions). As there are 5 cells which contain the value 0, we divide 197/120 by 5.

Remark: as the players have the choice to leave a coin or not, the sum of the "empirical" expected values is now different from 5. For example at step 1, it is equal to 2.88. It is interesting to notice that at step 2 under R2, a player who would like to use the behaviours of the others, should visit the cell which has been the less visited.

Expected number of coins left at step 1



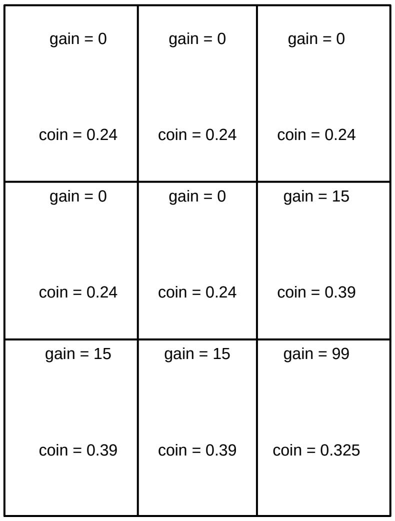
gain = 0
coin = 0.33
	gain = 0
	coin = 0.33
		gain = 0
		coin = 0.33

gain = 0
coin = 0.33
	gain = 0
	coin = 0.33
		gain = 15
		coin = 0.35

gain = 15
coin = 0.35
	gain = 15
	coin = 0.35
		gain = 99
		coin = 0.19

After the second step, we remark that the cell which contains 99 is still the one where we can find the smallest number of coins:

Expected number of coins left at step 2

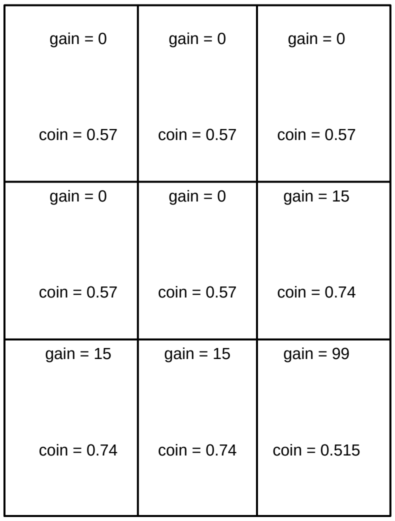


gain = 0
coin = 0.24
	gain = 0
	coin = 0.24
		gain = 0
		coin = 0.24

gain = 0
coin = 0.24
	gain = 0
	coin = 0.24
		gain = 15
		coin = 0.39

gain = 15
coin = 0.39
	gain = 15
	coin = 0.39
		gain = 99
		coin = 0.325

Information given after step 2



gain = 0
coin = 0.57
	gain = 0
	coin = 0.57
		gain = 0
		coin = 0.57

gain = 0
coin = 0.57
	gain = 0
	coin = 0.57
		gain = 15
		coin = 0.74

gain = 15
coin = 0.74
	gain = 15
	coin = 0.74
		gain = 99
		coin = 0.515

After step 3, the cell which contains 99 is now the one with the highest number of coins:

Expected number of coins left at step 3

gain = 0
coin = 0.18
	gain = 0
	coin = 0.18
		gain = 0
		coin = 0.18

gain = 0
coin = 0.18
	gain = 0
	coin = 0.18
		gain = 15
		coin = 0.32

gain = 15
coin = 0.32
	gain = 15
	coin = 0.32
		gain = 99
		coin = 0.67

Information given after step 3

gain = 0
coin = 0.75
	gain = 0
	coin = 0.75
		gain = 0
		coin = 0.75

gain = 0
coin = 0.75
	gain = 0
	coin = 0.75
		gain = 15
		coin = 1.06

gain = 15
coin = 1.06
	gain = 15
	coin = 1.06
		gain = 99
		coin = 1.185

After step 4, the cell which contains 99 is still the one with the highest number of coins:

Expected number of coins left at step 4

gain = 0
coin = 0.16
	gain = 0
	coin = 0.16
		gain = 0
		coin = 0.16

gain = 0
coin = 0.16
	gain = 0
	coin = 0.16
		gain = 15
		coin = 0.26

gain = 15
coin = 0.26
	gain = 15
	coin = 0.26
		gain = 99
		coin = 1.08

Information given after step 4

gain = 0
coin = 0.91
	gain = 0
	coin = 0.91
		gain = 0
		coin = 0.91

gain = 0
coin = 0.91
	gain = 0
	coin = 0.91
		gain = 15
		coin = 1.32

gain = 15
coin = 1.32
	gain = 15
	coin = 1.32
		gain = 99
		coin = 2.265

Conclusion: it seems that a player who played the less visited cell after step 1 and step 2 and the most visited cell after step 3 and step 4, would optimize his chance to find 99. We are now looking if this is really the case in our data.

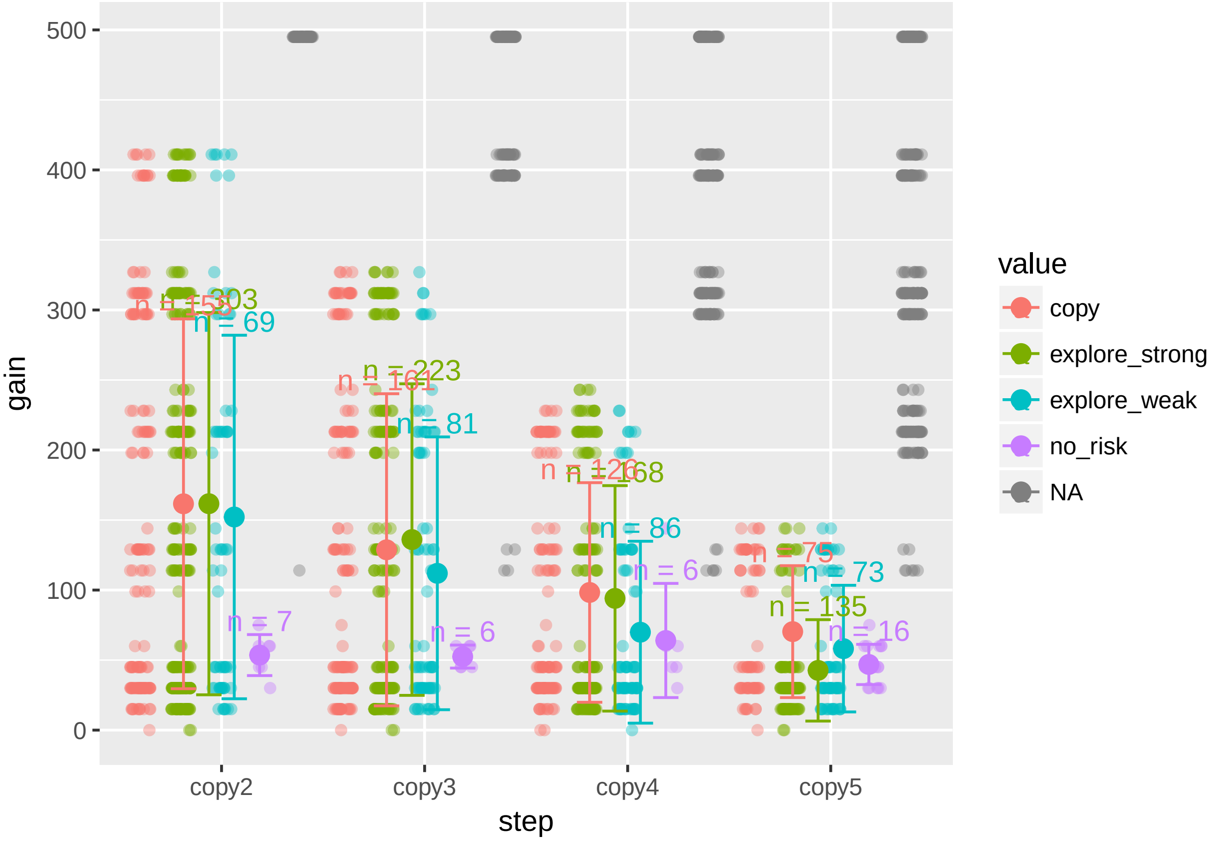
2.4.5 Behaviours among the players under R2

First, we will apply the previous function *cell_visited()* which allows us to see how often a cell has been visited per round.

For representing the data, we will tidy the data (codes presented in the .Rmd file).

We compute the mean and standard deviation obtained at each step and depending on the fact that a player explored new cases or not (codes presented in the .Rmd file).

We plot in y-axis the gain per round and in x-axis the steps. We represent in blue (resp. in red) the gain per round obtained when a player played a visited cell cell (resp. when he did not visit) knowing that the player had the opportunity to do it. We plot the gain per round in grey obtained by people who did not have the opportunity to visit (that means that they know where the cell 99 is located):



Interpretation: we can do the following remarks.

- at any step, the number of players who explore strongly is the most important. In other terms, players seem to play as if they did not trust the information given by others.
- at step 2, 3 and 4, there is no strategy which seems to give better results than the others.
- at step 5, it seems more interesting to copy rather doing something else.

The statistical test of comparasion of the means between the groups tends to give the same results : there are no significant differences at step 2 and 3. There is a small difference at step 4 and a significant difference at step 5.

```
##
##  Kruskal-Wallis rank sum test
##
## data:  gather_file_15_R2$gain[gather_file_15_R2$step == "copy2"] by factor(gather_file_15_R2$value[ga
## Kruskal-Wallis chi-squared = 1.598, df = 3, p-value = 0.6598

##
##  Kruskal-Wallis rank sum test
##
## data:  gather_file_15_R2$gain[gather_file_15_R2$step == "copy3"] by factor(gather_file_15_R2$value[ga
## Kruskal-Wallis chi-squared = 1.9466, df = 3, p-value = 0.5836

##
```

```
##  Kruskal-Wallis rank sum test
##
## data:  gather_file_15_R2$gain[gather_file_15_R2$step == "copy4"] by factor(gather_file_15_R2$value[ga
## Kruskal-Wallis chi-squared = 8.3655, df = 3, p-value = 0.03903
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  gather_file_15_R2$gain[gather_file_15_R2$step == "copy5"] by factor(gather_file_15_R2$value[ga
## Kruskal-Wallis chi-squared = 23.164, df = 3, p-value = 3.732e-05
```

2.4.6 The effect of copying/exploring stongly or weakly/re-playing 15 on the final gain

We are now looking for each player how does he behave during one session and trying to explain if its behaviours could explain the final gain.

As under R1, we look at four different behaviours : copying, exploring strongly, exploring weakly, taking no risk (codes presented in the .Rmd file).

At step 2: Most of players have decided to explore strongly at step 2 (29/40):

```
##
##           copy explore_strong   explore_weak
##             10             29              1
```

Besides, the average mean of the final gain obtained by the players who have a tendency to explore strongly (2802) is lower than the average mean of the group who has copied (3402). This could be explained by the fact that the players who copy at step 2 belong to some sessions where several players behave as "collaborators". In that case, we have that it is interesting to play the most visited cell

The final gain per group :

```
##           copy explore_strong   explore_weak
##       3402.300       2802.103       1920.000
```

At step 3: Most of players continued to explore strongly (22/40) at step 3 :

```
##
##           copy explore_strong   explore_weak
##             15             22              3
```

This time, the average mean of the final gain obtained by the players who had a tendency to explore strongly (2877) is slightly larger than the average mean of the group who have copied (2907). Players who have explored weakly have the highest score (3432).

The final gain per group :

```
##           copy explore_strong   explore_weak
##         2907.6         2877.0         3432.0
```

At step 4: Most of players continued to explore strongly (21/40) at step 4 :

```
##
##           copy explore_strong   explore_weak
##             15             21              4
```

Besides, the average mean of the final gain obtained by the players who have a tendency to explore strongly (2815) is slightly lower than the average mean of the group who copies (3025). Players who explore weakly have the highest score (3175).

The final gain per group :

```
##           copy explore_strong   explore_weak
##         3025.8         2815.0         3175.5
```

At step 5: Most of players have decided to explore strongly at step 3 :

```
##
##           copy explore_strong   explore_weak        no_risk
##             10             23              5              2
```

The final gain per group tend to show that there are no big differences between the groups:

```
##           copy explore_strong   explore_weak        no_risk
##       2900.700       2873.217       3250.800       2929.500
```

2.4.7 The effect of being a collaborator or a liar on the final gain

We are going to give some marks depending on the actions done by the players during one session. Players will get the following marks :

- +5 if they leave coin when they found 99
- +2 if they don't leave a coin when they found 0
- 0 if they don't leave a coin when they found 15
- 0 if they leave a coin when they found 15
- -3 if they don't leave a coin when they found 99
- -4 if they leave a coin when they found 0

We count the sum of good marks per player:

- A player with a score higher than 75 is a "collaborator"
- A player with a score lower than -75 is a "liar"
- Otherwise, they will be considered as "both"

Finally, we got the following statistics concerning the frequencies:

```
##
##         both collaborator         liar
##           16           10           14
```

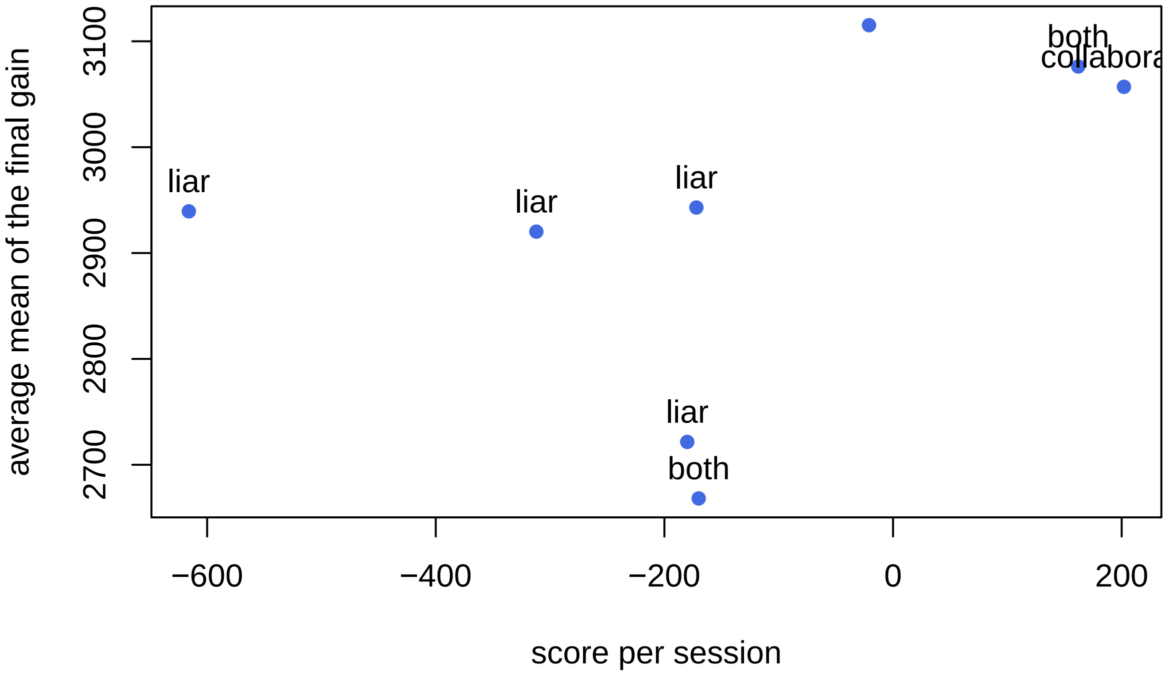
The final gains per group tend to show that the group of "liar" seems to get the higher gain:

```
##         both collaborator         liar
##     2793.938     2808.600     3172.500
```

2.4.8 The effects of being a collaborators/liars/both during a session

We have the intuition that the final gain obtained by a player during a session will depend on the number of players who belong to one of these categories. For example, if we consider a session where all people behave as "collaborators", we can think that they should use a stratgey such as described in the theoetical framework. However, if there are a majority of "liars", players will understand it and will change their strategy.

By using the same system of notation that previously, we give a score to a session which is supposed to indicate how the players behave in this session. We compare it to the average mean of final gain obtained in a session. We have also added in the scatter plot the identity of the winner of the session to check what was his profile:



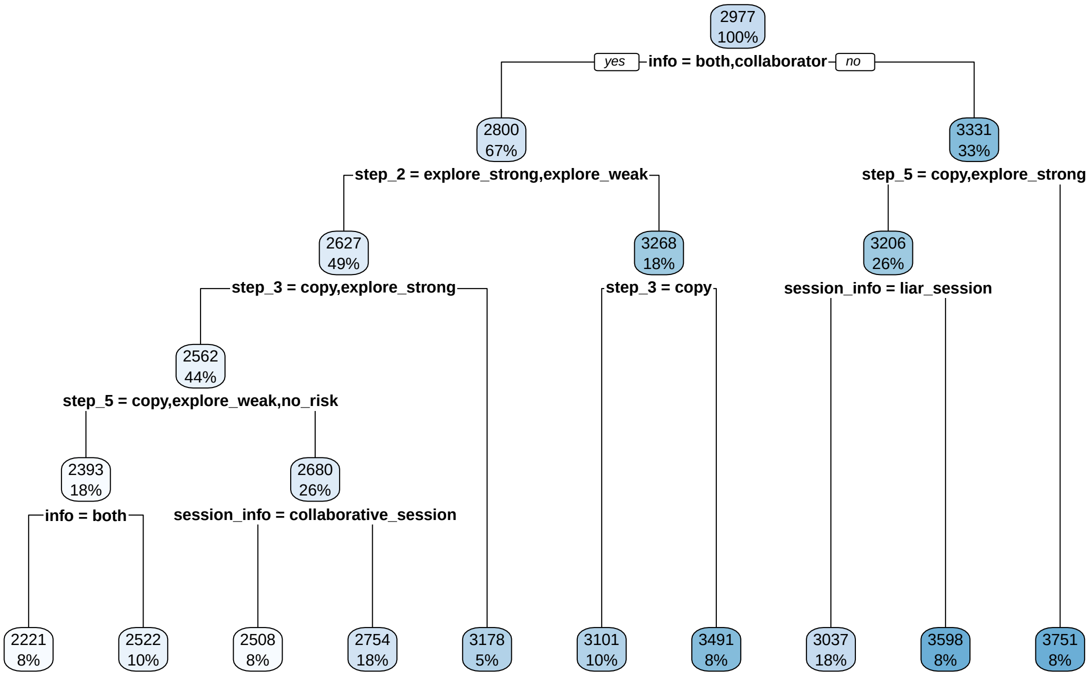
We observe that there are 3 sessions where players behave as liars (the sessions with a score lower than -100) and 4 sessions where they behave much more as collaborators (the sessions with a score higher than -100). In this last case, we can observe that the final gain is much more higher.

In the following session, we will also try to use this information related to the fact that a session has been done by players who collaborate or not.

2.4.9 Machine learning

We know that there is one player who behave badly (see first section). We delete him for the statistical analysis because they have a too strong influence.

Finally, we do a regression tree trying to explain the final gain obtained by players depending on their behaviours.


2977
100%
yes
info = both,collaborator
no
2800
67%
3331
33%
step_2 = explore_strong,explore_weak
step_5 = copy,explore_strong
2627
49%
3268
18%
3206
26%
step_3 = copy,explore_strong
step_3 = copy
session_info = liar_session
2562
44%
step_5 = copy,explore_weak,no_risk
2393
18%
2680
26%
info = both
session_info = collaborative_session
2221
8%
2522
10%
2508
8%
2754
18%
3178
5%
3101
10%
3491
8%
3037
18%
3598
8%
3751
8%

Interpretation: the variable which discriminates the most the players is the fact that player give good or bad information. The node with the higher final gain corresponds to players who are **liar** and do copy at the step 4.

2.4.10 Conclusion under R2

3 Statistical analysis of Stigmer 50

TBD