The contagion of luck

A simulation on gambling addiction

Marco Maurizio Disarò

6th February, 2016

1. Introduction

I want to build a model in NetLogo which simulates the action of gambling by different kinds of agents. Before illustrating further, it is important to establish who these agents are, in order to have a clear idea of what the model wants to achieve. Our gamblers can be distinguished on the basis of their gambling propensity, that is, the probability that the individual will decide to pay the lottery ticket if other hypothesis are satisfied, as it will more deeply described in next sections.

Then, the main idea which underlies the model is that there are some particular internal mechanisms which make a person more likely to spend his money on gambling. Many studies have been undertaken to understand them, with different sciences trying to explain this phenomenon according to their area of interest. Economists like Hicks (1965) and Pollack (1970) have observed that preferences for current consumption (included gambling) may be affected by past consumption levels. Then, a good can be considered *addictive* if its consumption has got some time-relation which links present with past (and even future). Nevertheless, these analysis don't preclude studies in other fields, like medicine or psychology, indeed, it is exactly the opposite. A mingling among different discipline may be the key to more deeply understand a complex phenomenon strictly linked with something as rarefied as human behaviour.

Of course, the model deals with individual choices, but the main topic is focused on how these choices can influence other people's decisions and, conversely, how they are modified by external behaviours. For this reason, this paper contains the word *contagion* in its title, to express how individuals are deeply affected by the rest of the society. Different social

compositions will lead to different global outcomes. In other words, this model tries to conciliate individual actions with an external framework which is determined by individuals themselves. Accordingly to this idea, the final results will show that people with high gambling propensity have a negative influence on near individuals, leading to a general increase in bets. On the other hand, it is also true that a huge presence of unwilling people have a positive effect on social health.

Another key point of the model is how people perceive external signals. This aspect of the question deals with subjective perception and elaboration of exogenous variables. In particular, I want to analyse how people perceive luck, even if it concerns with irrational beliefs, such as thinking that a store may be luckier since many people have won the lottery in last weeks there. It should be quite clear that this reasoning completely clashes with probabilistic theories and even the common sense, but nevertheless this is a fact. Guryan and Kearney (2009) have found out that stores which registered a win in state lottery experienced a significant increase in lottery consumption. Of course, part of this rise can be explained by a general boom in interest towards the game (e.g. media, grapevine), but this explaination doesn't cover the whole situation. Indeed, by observing the change in consumption of other stores, Guryan and Kearney found out that the lottery office where the winning ticket has been sold had registered a far higher increase in sales. They called this phenomenon *lucky store effect*, phrase wich perfectly express the deep belief that luck distribution is not as random as statistics wants us to believe.

The influence of individual perception is also present in the model through an increase in individual propensity to gambling in case of many wins in a store. This can be considered a similar kind of contagion with respect to the previous point, in such a way that people believes that luck is not just a matter of odds, but something different which can modify its spacial distribution.

Finally, I have already mentioned the dualism between negative and positive influence (assuming that gambling is a bad habit, of course). Then, if it is possibile to be led towards addiction, it is also true that external actions can help to reduce gambling consumption. One of the most thorny topics about statal responsibility is surely public intervention on this field, wondering if something has to be done in order to relieve a deep wound of society. Considering that governments are the biggest providers of lotteries, it may be difficult to imagine that such policies could be voluntarily adopted, since it would mean a reduction in revenues. Another option could be the implementation of consumer protections by the private stores themselves. Gambling is a huge sector, and many are the possibilities to risk your own money, for example, not just lotteries, but videoslots and *scratch and win* too. There are storekeepers who decide to get rid of any machines linked with gambling in order to help dangerously addicted people to reduce the waste of money in similar activities. In both cases (public policies or private intervention), a system of consumer protections is implemented in the model, in such a way that it is possible to reduce the average social gambling propensity and to stop the subtle contagion of luck.

2. The code

The first block of lines in my model deal with the creation of many breeds, both for turtles and links. These help to characterise the agents with their main attributes on the basis of their role in the simulation and they are helpful to distinguish different kinds of relations among turtles and to visualise alternative outcomes.

```
breed[compulsives compulsive]
breed[frequents frequent]
breed[normals normal]
breed[unwillings unwilling]
breed[stores store]

directed-link-breed [lottery-winnings lottery-winning]
directed-link-breed [lottery-losings lottery-losing]
directed-link-breed [protections protection]

globals [gamblers]
```

The opening five lines concern turtles, making it possible to distinguish between the agents who will gamble during the simulation and the static stores, which are the place where other turtles can gamble at. The names adopted for the four typologies of individuals are chosen to express their gambling propensity, with *compulsives* being the most likely to bet and *unwillings* the most risk averse. In order to streamline following steps, the globals variable *gamblers* has been included and, as it will be possible to see in the setup process, it contains the four breeds of agents, that is, all the turtles except for the stores.

Furthermore, there exist breeds for links too. *lottery-winnings* and *lottery-losings* have a quite straightgforward interpretation, as they are created each time a bet occurs. As I'll explain farther, they just have a computational purpose, as they never appear in the graphical interface. Conversely, *protections* can also be graphically visualised during simulation, besides having an important role in the code, of course.

The following lines describe the main attributes of all the elements appearing n the model, starting with gamblers, but stores and links too.

```
compulsives-own [propensity
                 wage
                 money-won
                 ]

frequents-own [propensity
               wage
               money-won
               ]

normals-own [propensity
             wage
             money-won
             ]

unwillings-own [propensity
                wage
                money-won
                ]


stores-own [luck]

protections-own [duration]
```

Agents are characterised by three main attributes, which are:

- *propensity*, which represents the probability that an agent will decide to bet if certain requirements are satisfied. It is a value between 0 and 1, and more risk-seeking individuals have a value closer to the upper bound. The decision to limit the values into interval [0;1] has been made to make a comparison with a percentage scale.
- *wage*, which indicates the money earned by the gamblers and its current budget without considering the money gained by gambling. This parameter is used to detect the percentage of agent who keep gambling even if they get into debt.
- *money-won*, which measures the winnings coming from bets. This attibute have an important role in contagion mechanism between agents.

On the other hand, stores own a unique attribute, which is *luck*. As *money-won*, it triggers reactions of agents influencing their propensity, but, differently from the previous one, it is specifically designed to represent what Guryan and Kearney have called *lucky store effect*. To distinguish lottery offices from gamblers, stores are given the shape of a grey house.

Finally, also links can have an attribute, but only if they belong to the breed *protections*. The assigned parameter is *duration*, which has the only purpouse of imposing a time limit to their existence. Indeed, these links are created each time an agent gamble in any stores, either if he wins or loses. If the number of links exceeds a certain value, we can imagine an external imposition ordered by government or due to private initiative such that agents can't gamble for a certain period of time, in order to reduce remarkable losses caused by compulsive behaviours. To measure this temporal aspect, the attribute *duration* is assigned to this kind of links.

Then it is time to setup. The process doesn't have any particular lines to note, with maybe only the last one being worthy of attention. It is needed to create the agentset *gamblers*, as already seen previously. It is defined as the set containing all the turtles except for the stores, which have a completely different purpose. It is not mandatory to group all four breeds from a structural point of view, but it significantly simplifies the code and reduces its length, since all agents shares same actions.

```
to setup
  ca
  reset-ticks
  setup-gamblers
  setup-stores

  set gamblers (turtle-set compulsives frequents normals unwillings)

end
```

In order to properly characterise them, the *setup-gamblers* command helps to define the initial propensity of each breed and the starting colour correlated with it. Red turtles correspond to risk-seeking gamblers, with propensity decreasing for yellow, green/lime and blue/sky ones. Nevertheless, the initial budget, defined using the slider in the interface window, is set equal for all agents, since there are no particular reasons to think that risk-averse individuals should have a different budget at the beginning. Indeed, starting from the same amount of money helps the analysis of the consequences generated by different behaviours.

Once the setup is ended, it is possible to continue with the *go* command, divided into four procedures.

```
to go

  tick
  adjust
  propension-indicator
  contagion
  gamble

end
```

The first two processes are used to make adjustments or changes at the beginning of each tick, while the last two represent the main structure of the model. Instead of following the order in which procedure are listed, I start by describing the fundamental action of gambling. Since it is a quite complex procedure, I split the block into smaller segments in order to highlight the most important issues.

```
to gamble

  ask gamblers [let nearest-store min-one-of stores [distance myself]
                  if distance nearest-store < 2
                    and random-float 1 < propensity
                      and count my-links with [breed = protections] < 1.1
```

The *gamble* procedure quite obviously concerns gamblers and, for this reason, the commands are asked to them. The first lines of the procedure needs as requirements to be fulfilled. In order to gamble, agents have to satisfy three hypothesis:

1. They have to be at a distance not greater than 2 from the closest store with respect to themselves. This particular store is identified thanks to the creation of the temporary variable *nearest-store*, whose name perfectly explains by itself how it is created. This mean that each gambler will have a unique (*min-one-of*) store as reference point to calculate the distance criterium.
2. A random number between 0 and 1 (*random-float 1*) is drawn and it has to be lower to the agent's propensity. The higher the propensity, the higher the probability that the agent will decide to gamble. This passage has to be interpreted as if an individual who walks close to a store wonders if it is appropriate to bet or just if he wants to do it.
3. The agent have to have no more than one single link with breed *protections*. The origin of these links can be found in the next lines of this procedure, then it makes sense to explain it after having reported them.

```
[ifelse implement-protections?

  [create-protection-to nearest-store [set color white
                                       set duration 5]
       ifelse show-protections?
       [ask protections [set hidden? false]]
       [ask protections [set hidden? true]]
   ]

  []
```

The creation of these links is conditional to an *ifelse* variable and, in particuar, to the activation of the switch *implement-protections?*, which can be selected in the right part of the interface window. Since there is an *ifelse* condition, two are the possible outcomes:

- the requirement is satisfied, that is, the switch is turned on. In this case, *protections* links are generated, but it doesn't mean that they can be seen. Indeed, the visibility is contioned to another *ifelse*, activated again by a switch (*show-protections*). I added this last option in order to leave the users the choice, since the graphical representation of links may be too much invasive and somebody may be interested only in the actual consequences of the implementation of this kind of policy.
- the switch is turned off. This means that protection policies will not be adopted during the simulation, making the *show-protections* switch definitively redundant. Since nothing occurs, the second bracket is empty, without having negative consequences on the rest of the code.

It is interesting to note that the third requirement at the beginning of the procedure (*count my-links with [breed = protections] < 1.1*) is important if we want to analyse the influence of restricting policies on gambling, but it doesn't produce any problem if *implement-protections* is turned off. Indeed, in this case the number of links with breed *protections* is always 0, so the condition is always fulfilled.

```
ifelse money-won > 0
  [set money-won money-won - 10]
  [set wage wage - 10]

    ifelse random-float 1 < probability-of-winning
        [if propensity < 1
          [set propensity propensity + 0.05
           set money-won money-won + 35
           create-lottery-winning-to nearest-store [set hidden? true]
          ]
         ]

        [if propensity > 0.05
          [set propensity propensity - 0.01
           create-lottery-losing-to nearest-store [set hidden? true]
          ]
         ]
```

Finally, there is the actual gamble. The first *ifelse* concerns the payment methods of the bet. If the agents owns a positive amount of winnings by gambling, it is more likely that she decides to use that capital to fund future bet, otherwise, she must take money from the sum earned by working. Whatever the situation is, the cost of a single bet is 10 units.

After having checked the requirements and having paid, the draw finally occurs. For this simulation, an instantaneous lottery is imagined, immediately knowing the result of the bet. The last *ifelse* condition deals with chance, since a randomly selected value between 0 and 1 is compared with the probability of winning chosen via the slider of the same name in the left part of the interface. Since there is an *ifelse*, there are two possible alternatives again:

- *random-float 1 < probability-of-winning* → the agent wins. Of course, the most immediate consequence is an increase in *money-own* fund, but a psychologic component must not be neglected. Many studies have tried to explain the rational and emotional reactions due to a

win. In order to reflect this phenomenon which combines self-accomplishment and excitement, lucky gamblers' propensity rises, making it more probable that the agent will play again in future periods. For acceptability reasons, this increase occurs only if propensity is not greater than 1, otherwise it wouldn't make sense to compare it with *random-float 1*, whose upper bound is 1 indeed.

- *random-float 1* < *probability-of-winning* → the agent loses. No money is won by the gambler and, as a consequence of failure, her propensity decreases. The important fact to note is that this reduction is smaller, in absoute value, than the increase in case of win (-0.01/+0.05). This discrepancy wants to reflect a phenomenon observed by many academics, that is, the prevalence of positive events against misfortunes. People tend to remember happy moments and to easily forget other events which didn't have positive outcomes.

In both cases, a link is generated between the agent and the store. Depending on the result of the bet, its breed is *lottery-winnings* and *lottery-losings*. Since they are important for a mere computational purpose, their attribute *hidden?* is set as *true*. Their role is explained in the last lines of the *gamble* procedure.

```
ask stores [set luck luck + count my-links with [breed = lottery-winnings]
                          - count my-links with [breed = lottery-losings] / 2]
```

Once all agents have decided whether playing or not, a new command is triggered, this time regarding stores. Links previously created are used to measure the change in percieved luck or, as already mentioned in this paper, the so-called *lucky store effect*. Moreover, for the same reason taken in consideration to explain different absolute values in propensity change in case of win and loss, also perception of good luck is overstated with respect to bad luck. Then, the positive influence due to a win is counterbalanced with the negative effect of two losses.

Now that the main procedure should be clear, it is easier to explain what the purpose of *adjust* is. This procedure is run at the beginning of every tick and its aim is correcting meaningless results, rearranging attributes and preparing the conditions needed for a new *go* process.

```
to adjust

  ask stores [if luck < 0
               [set luck 0]
             if luck > 10.5
               [set luck 10]
             ]

  ask gamblers [if money-won < 0
                 [set money-won 0]

                if money-won > 0
                  [set money-won money-won - 2
                   set wage wage + 2]
                ]

  ask links with [breed != protections] [die]

  ask protections [set duration duration - 1
                   if duration = 0 [die]
                  ]

end
```

The first lines are required to avoid that the stores' attribute *luck* assume negative or too much large values. These assumptions are not purely discretionary, since it is difficult to imagine that a store gains such a bad reputation that agents are discouraged to gamble there just because of some rumours. If a person wants to spend her money in gambling, surely it is not the case that she changes her action for this reason. The existence of an upper bound, besides having the computational purpose to avoid uncontrollable chain reactions, is useful to set a scale, a close interval for *luck* to easily compare stores' attraction.

Next step concerns gamblers and, in particular, their *money-won* attribute. Since its implementation regards subjective perspective of luck by external agents and the imitation mechanism that will be explained farther, negative values are not acceptable, otherwise it could measure misfortune too. Furthermore, since simulations allow to consider a time horizon, the total amount of money won by gambling cannot be computed just as a simple sum of incomes in different periods. Agents are encouraged to gamble if they meet another agent who has won a considerable amount in a limited period of time, not in her whole existence. For this reason, at the beginning of each tick, 2 units of *money-won* is converted into *wage*, indicating that this quantity is now considered as well-established wealth not linked with gambling.

Finally, also links need some adjustments. *lottery-winnings* and *lottery-losings* are no more useful, since they have already been used to modify stores' *luck* at the end of last tick and to plot a graphical representation of the cumulative number of wins and losses. Links with breed *protections* are the only exception. They are asked to decrease *duration* and to disappear only if this attribute reached the value 0 (*if duration = 0 [die]*). If this occurs, it does mean that a while has passed from the bet that generated that link, which means that storekeepers have no more reasons to prevent indidual by gambling again.

In addition to the previous changes, there are still two important modifications that concern with gamblers, but I preferred to split this processes because they have a quite important weight inside the code. That is, the following step is the *propensity-indicator* procedure.

```
to propensity-indicator

  ask gamblers [if propensity < 0.25 [set color sky]
                if propensity >= 0.25 and propensity < 0.5 [set color lime]
                if propensity >= 0.5 and propensity < 0.75 [set color yellow]
                if propensity >= 0.75 [set color red]
                ]

  ask gamblers [if wage + money-won < 0 [set shape "sheep"]
                if wage + money-won >= 0 [set shape "person"]
                ]

end
```

The main purpose of this lines is tracking the development of gamblers' behaviour. In particular, colours are used to indicate the propensity level of each individual, an expedient useful to have an instantaneous overview in the interface window. Looking at differently coloured turtles is way easier than checking the breed of all agents, making the interpretation much more immediate. Moreover, changing only the external aspect of gamblers doesn't modify the breed at which the turtles owned. This allows to analyse the propensity variation of any specific agent, just comparing her initial breed with her current colour.

The following lines concern again agents' appearance, but, this time, their shape. The quantity *wage + money-won* is computed and, if it is strictly negative, the gambler assumes the aspect of a sheep. Similarly to colours, this device helps to track the number of agents indebted because of gambling who, nevertheless, keep betting. Many researches have found out that this phenomenon is not as rare as commonly imagined, indeed, is relatively quite spread. There are many case of people who resort to debt, an evidence that stresses the sensitivity of gambling addiction.

Lastly, there is the procedure that tries to replicate relations between agents and the entire system of irrationals behaviour which characterise individual choices. Since the spatial dimension has a fundamental role, the procedure is called *contagion*.

```
to contagion

  ask gamblers [ifelse allow-imitation?
                 [let near-gamblers gamblers with [money-won >= 40 and distance myself < 2]
                  if count near-gamblers >= 1
                    [set propensity propensity + 0.05]
                  ]
                  []
                ]
```

Similarly to what happens with protections in *gamble*, even the imitation process is submitted to a switch, then to an *ifelse* condition. If *allow-imitation?* is turned on, a contagious mechanism is triggered. If there is at least one agent who won very much in last periods (*money-won >= 40*) and if she is near enough (*distance myself < 2*), then the observing gambler is encouraged to bet even more. This is due to a psychological effect which mingle envy with hope and the final result is an increase in gambling propensity. On the other hand, if one or both conditions are not satisfied, the infectious process does not occur (expressed in the code as the empty square brackets).

The last analysed mechanism in the code is quite complex in its form, since it is a concatenation of many *ifelse* conditions. Luckily, many passages are similar to ones in *gamble*, making it easier the comprehension of the structure.

```
ask gamblers [ifelse count my-links with [breed = protections] < 1.1
```

Just one line, but extremely important. It introduces the protection mechanism explained earlier in this section and, since it is conditioned to an *ifelse* command, two are the possible results. The following lines describes the scenario in case of policy activation.

```
[forward random-float 4 left random 360
       set wage wage + 0.5
       if propensity > 0.05 and random-float 1 < 0.5
           [set propensity propensity - 0.001]
  ]

]


end
```

These lines conclude the *contagion* procedure and its effects reflect on gamblers who are asked to stop betting. As a result, the agent decides to go away from the store where he gambled in last periods and her wage increases. This gain can be imagined as the money gained by

working instead of spending money in gambling. Moreover, this increase in wealth is combined with a detoxification effect, which consists in a reduction in propensity. Since it is not an easy task to help addicted people, this decrease is quite low in magnitude and it has positive effect only with a 50% probability (*random-float 1 < 0.5*).

```
[let lucky-stores stores with [luck > 8.1 and distance myself < 2]
 ifelse count lucky-stores >= 1

  [let rush-store min-one-of stores [distance myself]
     move-to rush-store
```

If instead protective safeguards are not taken, the following lines are designed to emulate the *lucky store effect*. As already mentioned, another *ifelse* condition is present and it asks if there exists at least one store with the attribute *luck* greater than 8.1 in a limited radius. If we consider luck adjustments as reported in the last lines of *gamble*, then the possible values such that the effect is activated are included in the interval [8.5, 10]. If there are no lucky stores nearby, the reaction is identical to the one following the protections activation (movement, increase in wage and chance of detoxification), otherwise the mechanism is triggered. The nearest store is detected with the creation of the temporary variable *rush-store*, whose name wants to reflect the urge felt as a consequence of this contagious phenomenon, and the agent immediately move towards it. The assumption is that, since the store is distant up to 2 units, it is possible to instantaneously reach the store.

```
ifelse implement-protections?
   [create-protection-to rush-store [set color white
                                     set duration 5]
    ]
   []

        ifelse show-protections?
         [ask protections [set hidden? false]]
         [ask protections [set hidden? true]]

   ifelse money-won > 10
     [set money-won money-won - 15]
     [set wage wage - 15]

       ifelse random-float 1 < probability-of-winning
           [if propensity < 1
             [set propensity propensity + 0.05]
            set money-won money-won + 35
            create-lottery-winning-to rush-store [set hidden? true]
            ]

            [if propensity > 0.05
              [set propensity propensity - 0.005
               create-lottery-losing-to rush-store [set hidden? true]
               ]
             ]
```

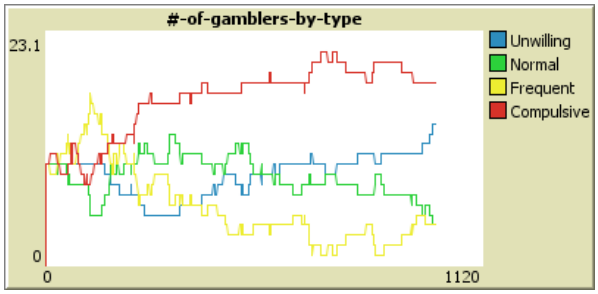
Once the agent reaches the store, the gambling process is almost identical to *gamble* one. The main difference is that the requirement *random-float 1 < propensity* is dropped, hence the contagion indiscriminately hits every breed of agents. Moreover, the cost of a ticket is greater thanthe one in the previous case, since it includes the extra-cost for the swift transport to the nearest store. Gamblers are in an overexcitement status and they are willing to suffer higher payments. For the same reason, the decrease in propensity due to a loss in absolute terms is lower than before (-0.005 instead of –0.01).

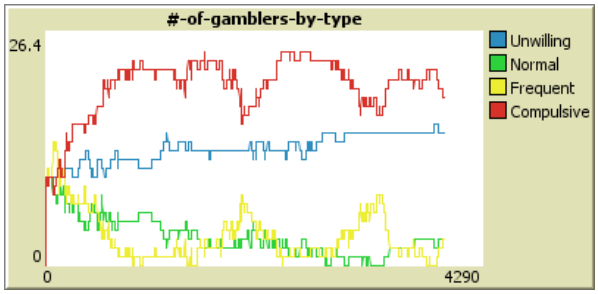
3. Experiments

In this section, the model is used to simulate outcomes generated by situations with different initial settings, varying parameters like the number of agents, the winning probability or the activation of particular mechanisms. If something else is not specified, the default values are assumed as:

- *#-of-compulsive-gamblers* = 10
- *#-of-frequent-gamblers* = 10
- *#-of-normal-gamblers* = 10
- *#-of-unwilling-gamblers* = 10
- *initial-budget* = 50
- *#-of-stores* = 10
- *probability-of-winning* = 0.250
- *implement-protections?* = Off
- *allow-imitation?* = On

Therefore, let's try to run the model using these default settings.

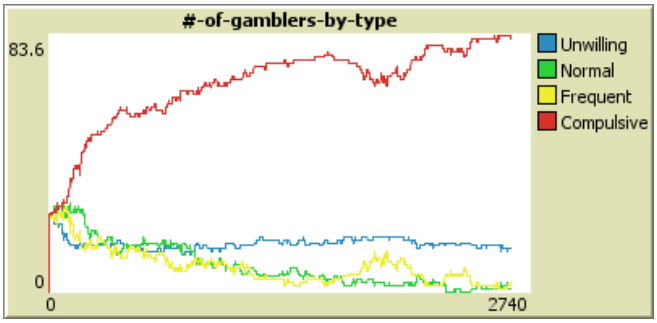




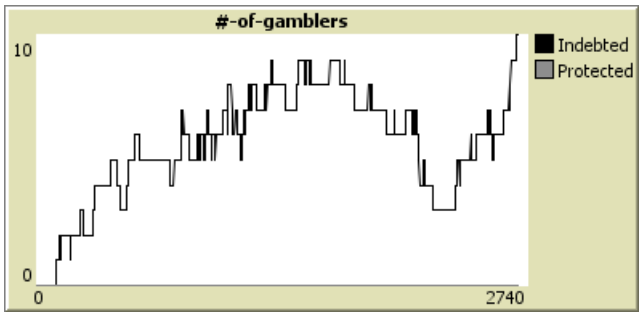
The picture on the left shows the composition of the population distinguished for gambling propensity in a time horizon of 1000 ticks, while the one on the right considers about 4000 periods. In both cases, the colours used are the same ones visible in the interface window, making it easier to interpret data. During the very first iterations, the four quotas have an irregular path, with curves intersecating many times and being around their initial value (10 for each breed). After 200-300 ticks, lines tend to acquire a regular trend and to mantain it in time. The most notable result is a relevant increase in the percentage of compulsive gamblers, overdoubling the starting value. Conversely, both frequent and normal agents tend to decrease in number, being around few units in the long horizon (with even periods without any of them). As a result, also the unwilling gamblers' percentage rises, but at a lower rate than *compulsives*. Nevertheless, with the flow of time, this gap tends to reduce and, even if red line still tops the blue one, the two curves seems to periodically converge. These first results show that agents tend to have clear ideas on their preferences. Average-propensity gamblers disappear and a choice is taken in favor of one of the extreme behaviours.

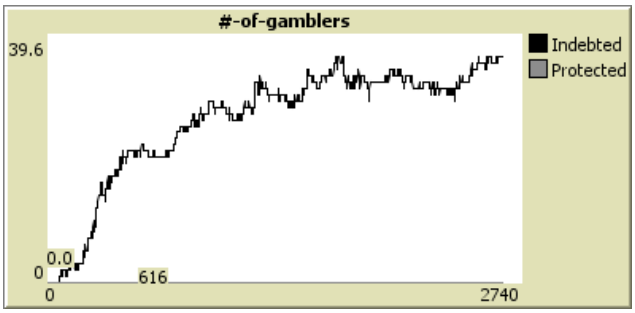
3.1 Does the number of agents matter?

In this experiment, I want to test if the size of the population has some effects on its propensity profile. The number of agents has been increased up to 25 per breed, then maintaining the same proportions in social composition. The size of population has passed from 40 to 100, value that makes it possible to obtain percentages in a straightforward way.

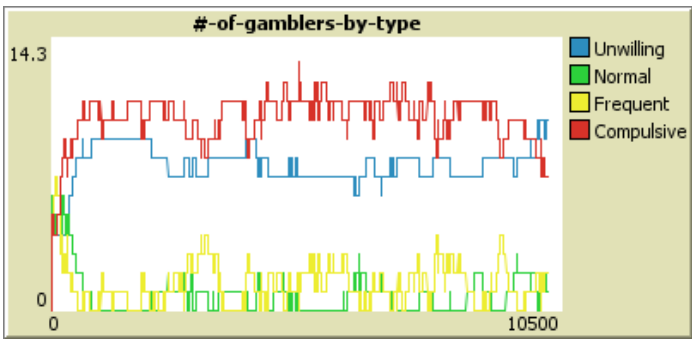


The final result is easily interpretable by looking at the plot above. Compared to the standard situation, there is a massive presence of compulsive gamblers, representing over the 80% of population (instead of the usual 50%). The most plausible explaination is that the high concentration of crowd makes contagion easier, leading to a dangerous chain reaction which affect almost the whole population.



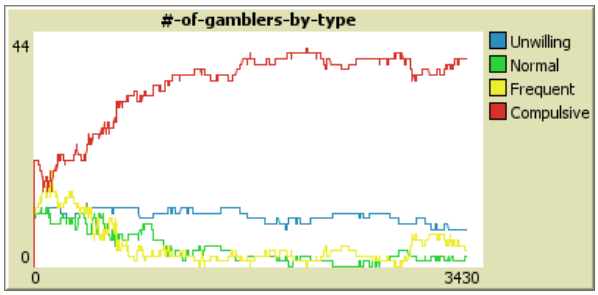


A similar result can be also found by analysing the graphs which concern the number of agents indebted because of gambling. The first image shows the outcomes of the default condition, while the second one represents the 100-agents case. In the first simulation, the percentage of indebted gambers never exceeds 25%, while in the latter, it almost reaches 40%. The cause of this phenomenon is the same one mentioned before. More agents mean that there is a higher probability that someone is going to have a negative influence on others' behaviour. On the other hand, for less than 10 agents per breed (in the graph below, they are 5), the path followed by the curves is way more similar to default outcomes.

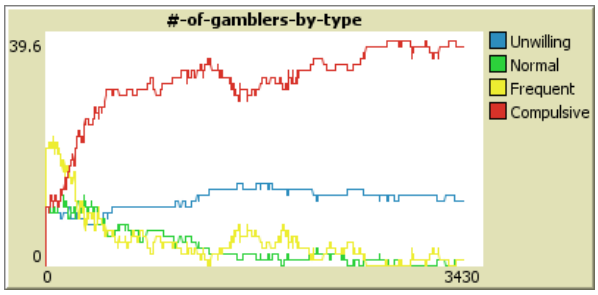


3.2 Does population's composition matter?

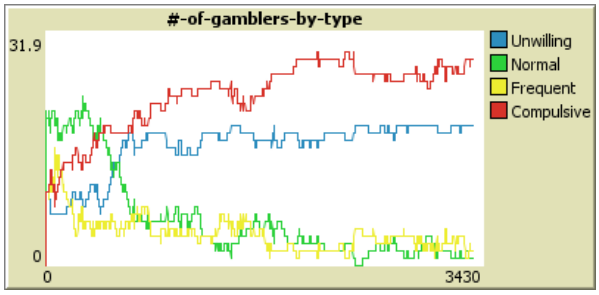
This experiment is thought to show how long-term trends change for population charachterised by an imbalance in composition. In all the four following examples, agentset *gamblers* consist of 50 units, 10 per breed except for one of the four, enlarged to 20 once each.



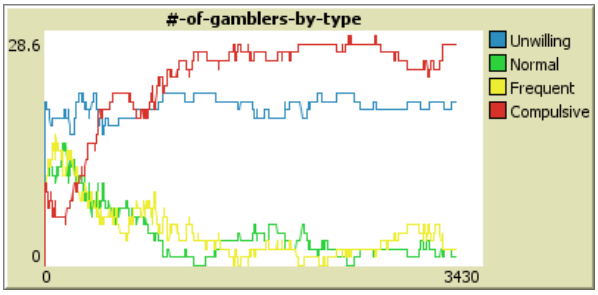
#-of-compusive-gamblers = 20



#-of-frequent-gamblers = 20



#-of-normal-gamblers = 20



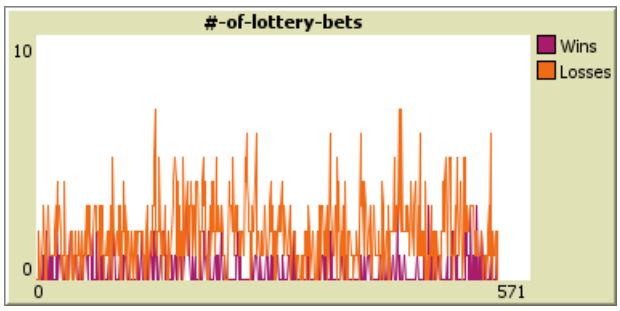
#-of-unwilling-gamblers = 20

In *frequent* and *normal* cases, corresponding curves experience a deep dive in both situations, corroborating the theory that middle gamblers tend to diverge to one of the two more radical behaviours. The key point is the one concerning the gap between *compulsives* and *unwillings*. The greater the propensity of the extended breed, the greater the gap between the blue and red curves. More risk-averse agents soften the natural vehemence of compulsive gamblers on social health.

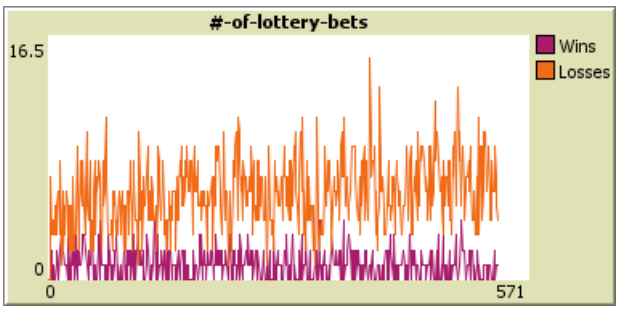
3.3 Does the number of lottery stores matter?

The model is not only composed of gamblers, indeed there are stores too. Until now, change in parameters have regarded only agents, but what does it happen when the population remains the same and the distribution of stores changes?

As it can be observed in the first two plots, if *#-of-stores* pass from 10 to 25, the amount of bets has an evident increase. The number of wins and losses are approximately doubled and it does make sense. A widespread distribution of lottery offices increases the chances that an agent is close enough to one of them to encourage her to gamble (via the *gamble* procedure).

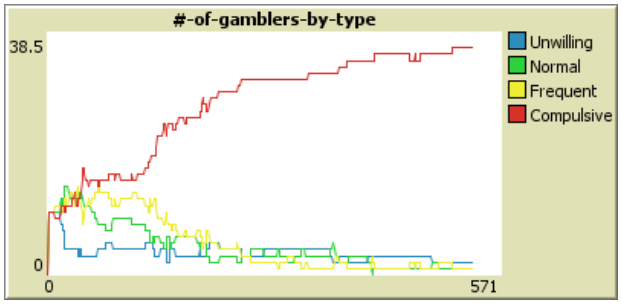


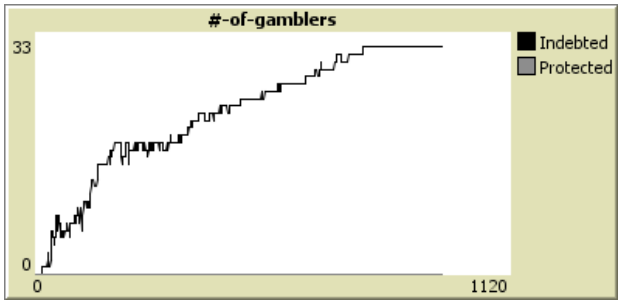
#-of-stores = 10



#-of-stores = 25

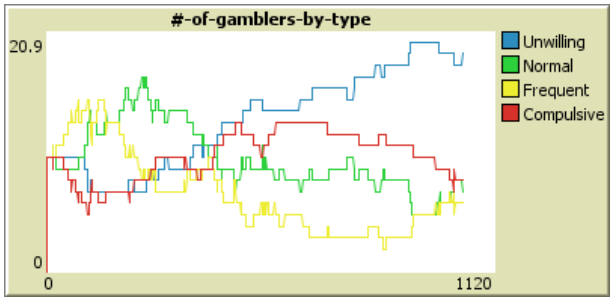
But what about propensity? The result is quite astonishing, since almost the whole population belongs to the compulsive fraction. The manic response to an increase in lottery supply rises serious issues about gambling addiction and its potential catastrophic effect. It is not surprising that in this experiment over 75 % of agents has a negative balance with huge amounts of debt after 1000 periods.

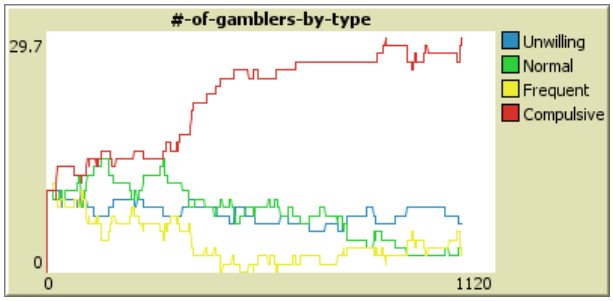




3.4 Does probability of lottery winning matter?

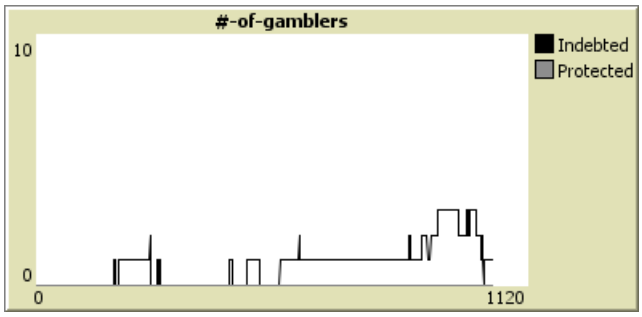
Since the model deals with chance, maybe the most immediate experiment concerns the variance of winning probability. The odds of a lottery are carefully shaped in order to generate the maximal revenues for the provider (governments, most of the cases). The default setting starts with a winning-probability of 25% and the foowing graphs show results, respectiely, for 20% and 30%.

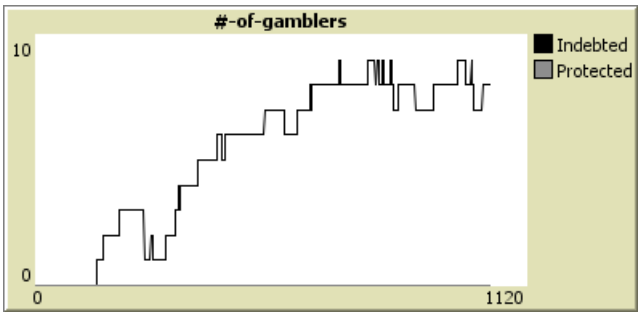




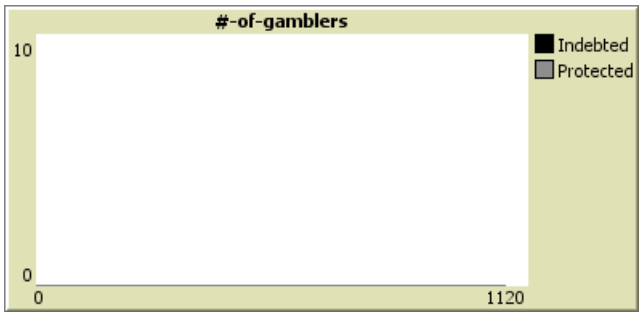
The results are quite clear. Even a relatively small variation in odds produces heavy consequences on agents' propensity. If it is easier to win, individuals are encouraged to bet and this perception translates into an average rise in social propensity. Moreover, the 20% example is one of the few cases in which *frequents* and *normals* hold a significant percentage in population's composition, reaching a level very similar to compulsive gamblers' one.

But the most interesting result is about the number of indebted agents. Intuitively, a greater probability of winning should provoke an increase in winnings, then the amount of debtors should decrease. The evidence shows exactly the opposite scenario.





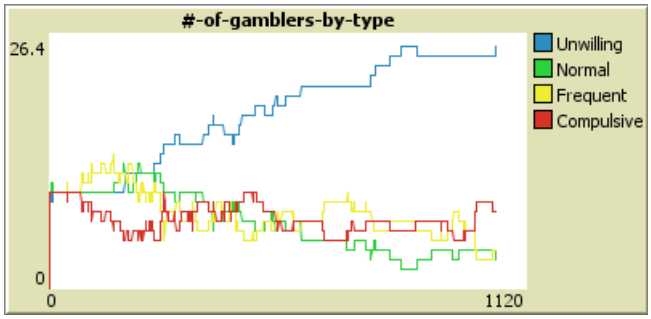
A possible explaination pass through propensity. We have already observed a rise in agents' will to gamble and, since the odds are more favorable, this translates into an increase in average *money-won*. This variation triggers the propensity contagion, hence people starts betting even more. The mania produces a chain reaction that, sooner or later, brings to an unstoppable mechanism with negative consequences. As a proof of this theory, it is sufficient to switch off *allow-imitation?*. The outcome confirms this process, since the are no indebted gamblers, indeed, they have conspicuous budgets thanks to the higher probability of winning (this time not counterbalanced by imitative mechanism).



probability-of-winning = 0.300 and *allow-imitation?* = Off

3.5 The imitation gamble

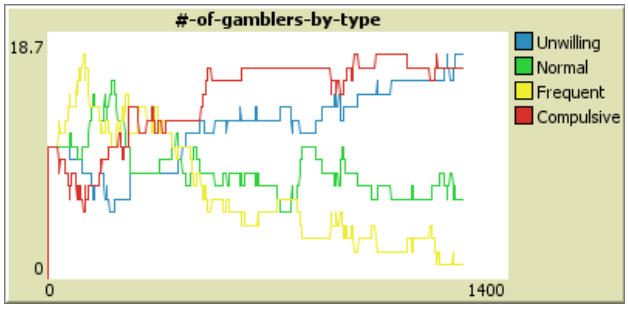
In the previous experiment, we have seen how the imitative process can have a great influence on final results, indeed, it seems to be even the most important aspect to consider in a model which tries to replicate gambling preferences. As before, let's turn *allow-imitation?* off.

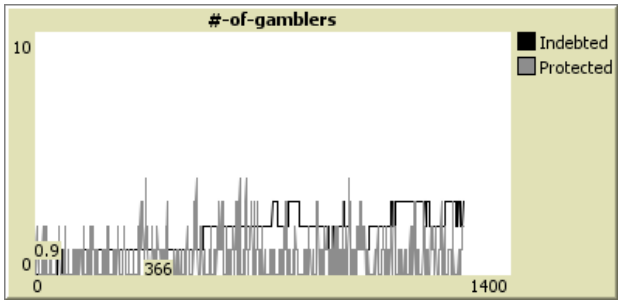


The outcomes shows an evident predominance of unwilling gamblers and it shouldn't be so surprising, since the importance and the influence of this mechanism has been stressed many times during this paper. In light of these evidence, it seems more and more plausible to explain gambling addiction as strictly related to similar psychological factors. More detailed studies could find effective policies to reduce bets and gambling disorders. Indeed, the model suggests that the number of transactions could significantly reduce by counterattacking imitative mechanisms.

3.6 A policy experiment

This last experiment is shaped in order to find a possible and applicable policy to limit gambling addictions. It is implemented by using the *protections* links, which are generated once per bet. The idea is the realisation of a plan with the purpose of protecting particularly sensitive individuals who have to manage a gambling disorder. It can be either a national program or a grassroots initiative promoted by enlightened storekeepers, but the mechanism is the same in both cases. People should not be able to spend their money in gambling without judgement, that is, if the individual can't control her behaviour, she is asked to quit the store after a certain number of bets. A practical way to implement this policy could be the creation of a magnetic card that must be swept before being able to bet. In this way, it could be possible to have a log of gamblers' routine in order to check the lie of the land and prevent them by having self-destructive behaviours.





The results are quite encouraging, maybe not incredible, but still it is a good starting point. The graph show positive variations in both number of protected individuals (grey line) and average gambling propensity. Indeed, the gap between *unwillings* and *compulsives* has remarkably reduced (in certain periods, it is even in favor of the former). Moreover, also *normals* and *frequents* values are significantly different from 0, showing a more balanced composition of population (maybe thanks to a better awareness about the problem).

Of course, it is not easy to implement such a policy, surely for initial administrative costs and investments, but also because of potential political pressure. Lottery and gambling are still an important source of revenues for both storekeepers and governments, then, if not impossible, protective policies seems to be at least very difficult to implement. Nevertheless, further analysis of gambling behaviours need to be encouraged and promoted in order to eradicate an illness of society like gambling addiction is.