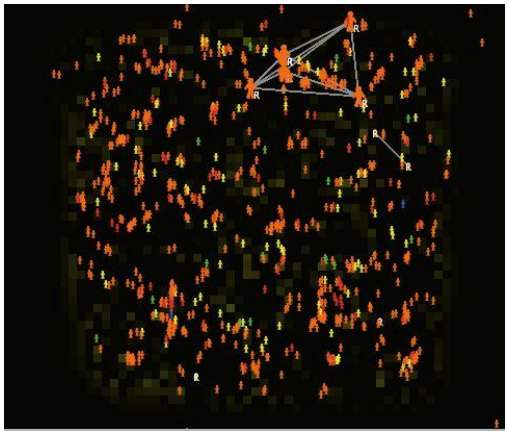
**Simulation Models For Economics – Year 2012/2013**

**FINAL PROJECT**

# *Wealth, Taxation and Elections*

**Luca Cesare Biagio Alasio**



***Abstract***

*The attempt to study economic complexity requires flexible models which let us observe distinct human aspects at the same time.*
*An economic model of a simplified society should include agents able to find, consume and store resources (grain) coming from the environment. I was interested in understanding how wealth distribution and resource availability are related and how they evolve.*
*In addition, I built an elementary democratic system in which some representatives can change three different taxation levels. Taxation income is transferred to the poorest creating a primitive redistribution system.*
*I thought it could be engaging to study how redistribution, taxation and elections affect the economy of such agents.*

## Introduction

I first approached simulation models a couple of years ago, during a lecture I attended in the Scuola di Sudi Superiori of Torino. In these years there have been several situations in which agent-based models proved to be an unavoidable and excellent instrument in understanding different complex systems and some of the many problems they raise.
In this work I used NetLogo in order to simulate a "small democratic society" with limited resources; this model might represent a further development of some basic ideas coming from the well-known world of Sugarscape.

*Main features*

Our space is a bidimensional toroidal space divided in patches; two sliders allow the observer to choose a rectangle in which resources will be present. In thisd case we have only one resource, called “grain”, and its amount varies in spece.

Turtles collect grain, store it, and consume a certain part of their resources in each time unit. They have a limited “vision”, in fact they notice the presence of grain only if it is not too far, furthermore they are able to move towards the patch with the highest grain quantity they can see.

Grin income, storage and consumption are liable for taxation.

Besides such basic skills, all turtles are required to choose a number of representatives proportional to their number.

Voting for a particular candidate id determined by the difference between agent’s and candidate’s preferences about taxation levels.

Elections consist of two steps: the creation of candidates and the choice of representatives; a switch allows to decide whether reelection is possible or not.

Representatives can form alliances, represented by links, on the basis of their preferences in the field of taxation. The largest alliance becomes the majority and it will be able to vary taxation levels.

The observer can channel taxation income into grain redistribution for the poorest agents’ benefit.

It is possible to choose if agents are subject to death caused by absence of grain or by old age; if so, and if the agents’ wealth crosses a certain threshold, they can reproduce and they have share their current grain amount with the new agent.

*Interpretation*

The model might seem quite complicated since it is meant to simulate two different social phenomena: a population evolution and an essential democratic system but I think that if the observer has already been introduced to NetLogo, he or she should not have difficulties in understanding how it works.

I have made several simplification, as well as unrealistic hypothesis, such as the lack of public administration or the assumption that all agents have similar goals but, even so, interesting aspects have emerged.

Wealth distribution is evidently asymmetric in most of parameter configurations, when the “death” switch is activated lower classes are extremely larger than upper classes, while such disproportion is mitigated when turtles do not die nor reproduce.

Redistribution effects are not always as conspicuous as we could expect but, in specific conditions, its introduction sometimes determinates a population explosion. Other results and some experiments will be presented in detail in a later paragraph.

I would like to thank Professor Pietro Terna for his great help and his constant support.

**Model description**

*Patches*

```
patches-own
[ grain-here
  max-grain-here ]
```

There are two variables related to patches, grain-here reports the current amount of grain present on each patch, while max-grain here is the maximum amount of grain that a patch can contain. The first variable changes in each time unit, while the second one is determined (up to a random variable) only at the beginning of every simulation.

```
to setup-patches
  ask patches
    [ set max-grain-here 0
     if (abs(pxcor) < X and abs(pycor) < Y)
     [
      if (random-float 100.0) <= percent-best-land
      [
        set max-grain-here max-grain
        set grain-here max-grain-here

        if (pxcor > X / 2 and pycor > Y / 2)
        [ set max-grain-here max-grain * 3
          set grain-here max-grain-here ]
       if (pxcor < (-1 * X / 2) and pycor < (-1 * Y / 2))
        [ set max-grain-here max-grain / 3
          set grain-here max-grain-here ]
    ]]]
```

Patches whose coordinates are smaller (in absolute value) of X and Y will contain grain, the other won't. Grain is not uniformly distributed: it is abundant in the upper right corner of the world and it is lacking in the opposite corner.

```
  repeat 5
  [ ask patches with [max-grain-here != 0]
        [ set grain-here max-grain-here ]
      diffuse grain-here 0.25 ]
  repeat diffusion
    [ diffuse grain-here 0.25 ]
  ask patches
    [ set grain-here floor grain-here
      set max-grain-here grain-here
      recolor-patch ]
end
```

In order to have a gradual variation of grain quantity, patches are asked to diffuse their grain; the process is repeated five times so that the result is

(aesthetically) better. The intensity of patches color is proportional to their grain abundance.

*Turtles*

```
turtles-own
[ age life-expectancy metabolism vision children
  wealth-tax-pref income-tax-pref consumption-tax-pref
  candidate representative num-votes abstention tolerance
  income wealth consumption ]

to set-initial-turtle-vars
  set age 0
  face one-of neighbors4
  set life-expectancy life-expectancy-min +
            random (life-expectancy-max - life-expectancy-min + 1)
  set metabolism metabolism-min + random metabolism-max
  set wealth metabolism + random 50
  set vision 1 + random max-vision
  set children 0
end
```

If the death switch is on, turtles grow in age, consume grain and con reproduce (with a sort of mitosis), age, predetermined life expectancy, grain consumption (metabolism) and number of children is saved in two corresponding variables.
Every turtle has a non-negative grain income and consumption (for the sake of simplicity, the last one is equal to metabolism). Obviously, Income is added to wealth while consumption is subtracted, which means wealth might be negative (if turtles can't die).

```
to setup-voting-system
  set candidates no-turtles
  set representatives no-turtles
  ask turtles
  [ set wealth-tax-pref abs random-normal 0 0.01
    set income-tax-pref abs random-normal 0.3 0.1
    set consumption-tax-pref abs random-normal 0.2 0.1
    set candidate false
    set representative false
    set num-votes 0
    set abstention 0
    set tolerance random-float init-tolerance ]
```

"Candidate" and "representative" are true if the considered turtle belongs to the corresponding agentset. Turtles identify a set of candidates whose three taxation preferences differ less than "tolerance" from their own. If such set is empty, "abstention" becomes true, otherwise they chose randomly one of their candidates and make his number of votes increase.

*Links*

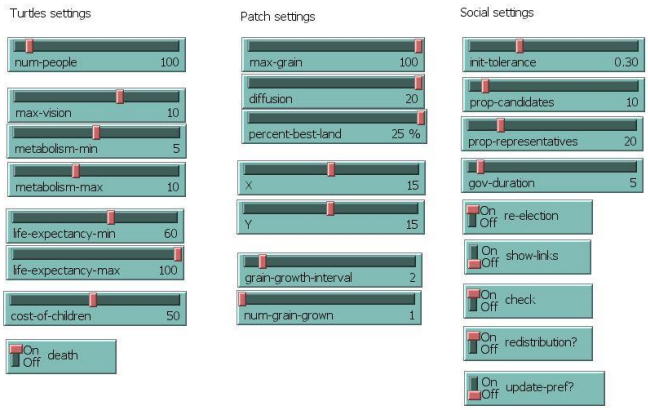
Links represent alliances between representatives. First representatives with similar preferences create a link, then all turtles connected in some way create

new links with each other (complete graph). This last operation is repeated by the largest alliance in order to form the majority.

*Globals and interface*

```
globals
[  tax-income wealth-tax income-tax consumption-tax
   candidates representatives majority
   wealth0 income0 consumption0  ]
```

Taxation variables and election-related agentsets are represented by global variables and are updated regularly; “wealth0”, etc. can be interpreted as artificial poverty levels: if a turtle’s values get under them, the turtle is suitable for redistribution. In our case such levels are fractions of the respective mean value.




The interface is divided into three groups of operating controls, which govern, respectively, turtles, patches and social behavior.

Two important regulations concern the already mentioned death switch and the “check” switch: the first one strongly influences the evolution of the model and the second one ensures that taxation levels make sense (between 1 and 99 per cent).

*Model working*

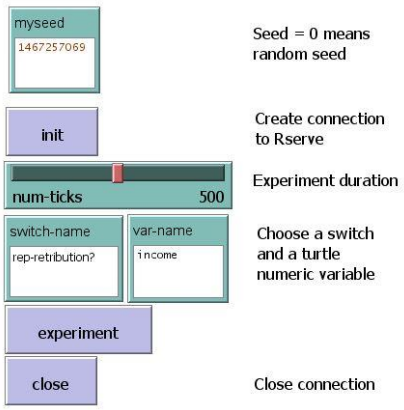
```
to go
  ask turtles [ turn-towards-grain ]
  simulate-economy
  simulate-gov
  ask turtles [ move-eat-age-die ]
  recolor-turtles
  if ticks mod grain-growth-interval + floor (abs(random-normal 0
2))=0 [ ask patches [ grow-grain ] ]
  simulate-elections
  ifelse show-links = false
   [ask links [ set hidden? true ]]
   [ask links [ set hidden? false ]]
  check-values
  tick
```

```
  my-update-plots
end
```

The “go” procedure is the core of the model. First of all it makes turtles move towards the best source of grain. When different turtles reach the same patch, they divide up the grain available. The procedure “go-towards-grain” is very similar to the one used in the NetLogo Library model “Wealth Distribution”.
The procedure “update-economy” updates incomes and wealths, implements taxation and redistribution, while “simulate-gov” allows majority to modify taxation levels and updates redistribution levels (“wealth0” ,…).
“recolor-turtles” enables the observer to recognize turtles of different classes at a glance, since they are colored according to their wealth class.
The democratic system is regulated by “simulate-elections”: candidates are chosen randomly, reelection can be forbidden by the observer as well as government duration. The number of representatives is proportional to the population number by means of a constant fixed by the observer.

**Experimental results**

I was interested in understanding how model evolution is affected by activation or deactivation of certain switches. I tried to take advantage of the software R in order to have a quite precise answer to question such as “Does wealth distribution actually depend on  redistribution in this model?”.
In this setting reproducibility is essential, in fact it is possible to choose the seed used by NetLogo during experiments thanks to a dedicated input box (notice that in this case seed = 0 means random seed).
In order to produce nice statistical plots NetLogo sends data to R via the Rserve extension (http://jasss.soc.surrey.ac.uk/15/3/8.html).

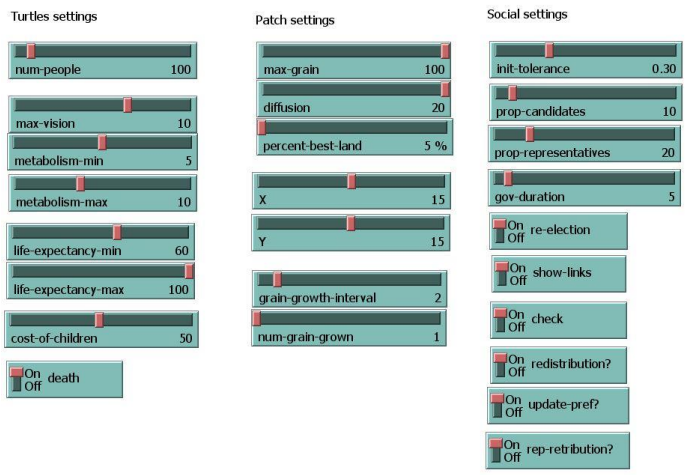



Several experiments can be conducted from the interface: after creating a connection between NetLogo and R (“init”), the observer should write in two input boxes the name of the desired switch and turtle variable.
Pressing “experiment” button we get a table consisting of four plot: the first column is referred to the “off” state of the switch, the second one to the “on” state. The upper plots contain time on the horizontal axis and the evolution of the mean of our variable on the vertical one; also the regression line of these data is drawn. The underlying plots are “boxplot” of the previous sequences.

```
to run-exp [numrep n]
    setup
```

```
    let var []
    set var [runresult(var-name)] of turtles
    rserve:put "x" mean var
    rserve:put "t" ticks
    rserve:eval "df1 <- data.frame(variable = x)"
    rserve:eval "df2 <- data.frame(time = t)"
    repeat numrep
      [ go
        set var [runresult(var-name)] of turtles
        rserve:put "x" mean var
        rserve:put "t" ticks
        rserve:eval "df1 <- rbind(df1, x)"
        rserve:eval "df2 <- rbind(df2, t)"
       ]
        rserve:eval "df <- cbind(df1, df2)"
        rserve:eval "attach(df)"
        rserve:eval "regr <- lm(variable~time)"
        if n = 1
        [rserve:eval "par(mfcol=c(2,2))"]
        rserve:eval "plot(time, variable)"
        rserve:eval "abline(regr)"
        rserve:eval "boxplot(variable)"
end
```

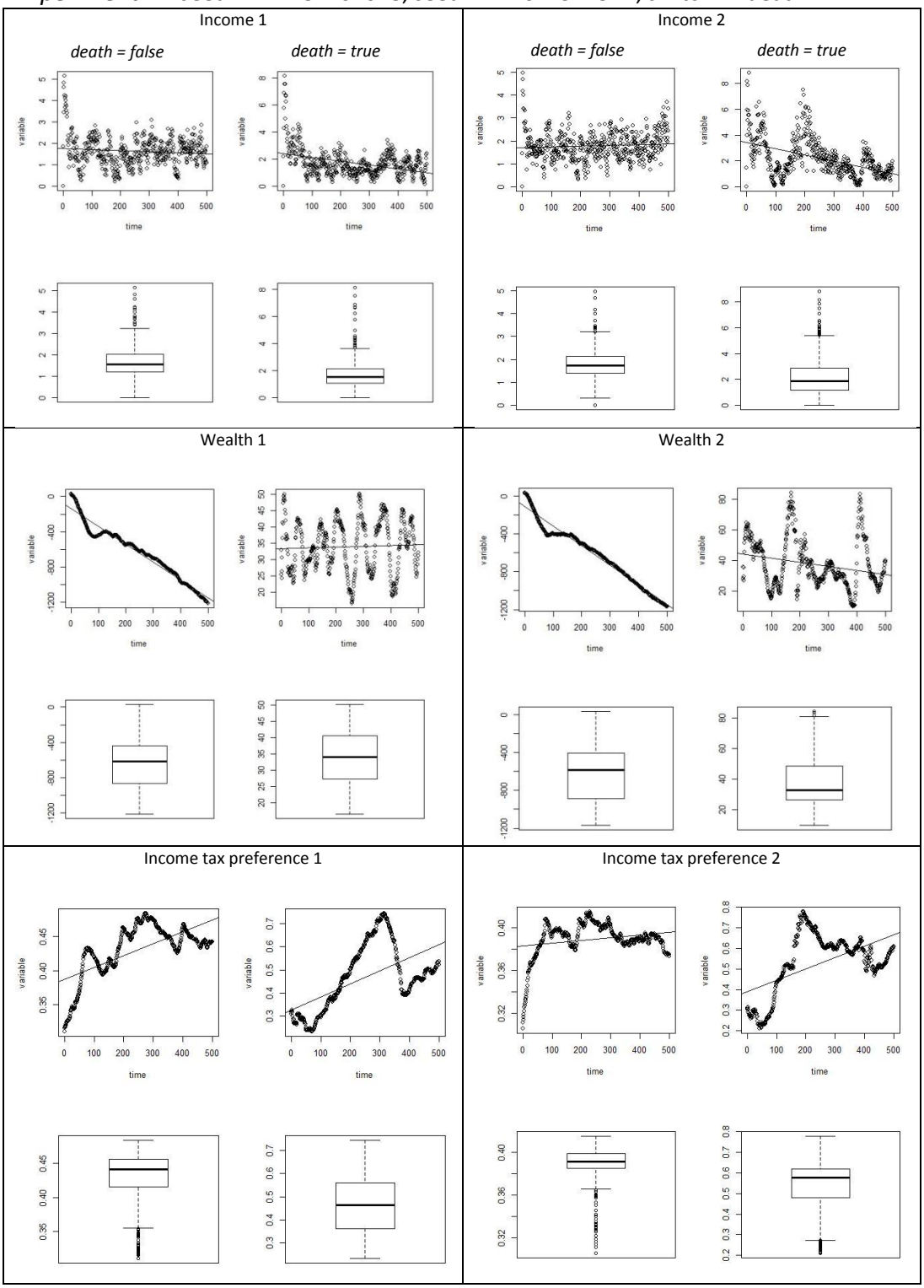
I used commands “run” and “runresult” in order to make Netlog transform input strings into variables, then I sent data to R and put them in the form of “data frames”, finally I used “lm” (linear model) to perform linear regression.




Settings for experiments 1 and 2

In the following pages I will briefly discuss some simple results and plots showing the model on the job.
In order for these experiments to acquire statistical relevance a lot of observations should be made.
I have restricted to report just some naïf examples of investigation which could be examined in depth in other projects.

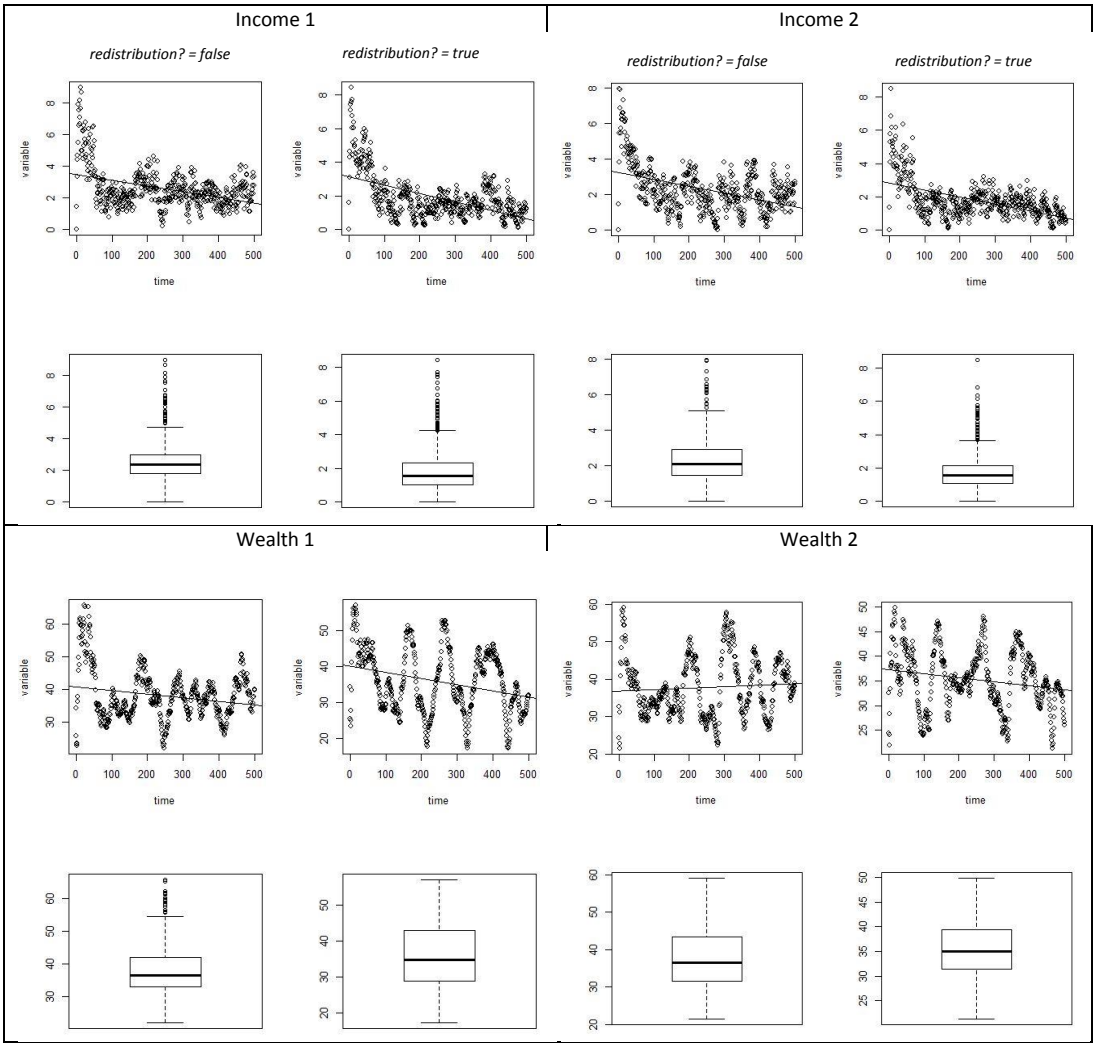
*Experiment 1* - seed 1 = -1187762028, seed 2 = -2021941971, switch = “death”

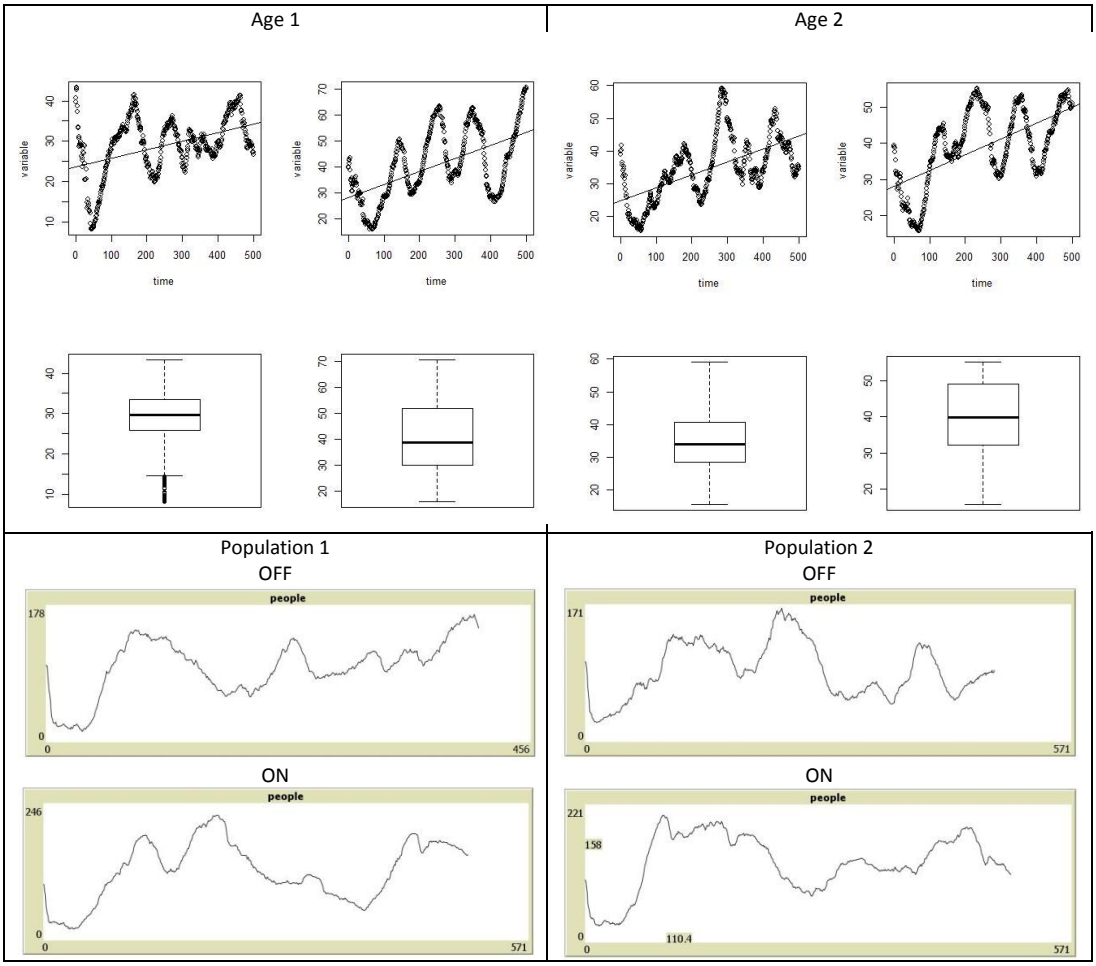



It seems death doesn’t have a strong effect in incomes, even if they might decrease a bit faster when death is on. Wealth behavior is very different: without

death there is no evolution and high metabolism. Income tax preference seems to be affected in terms of variance.

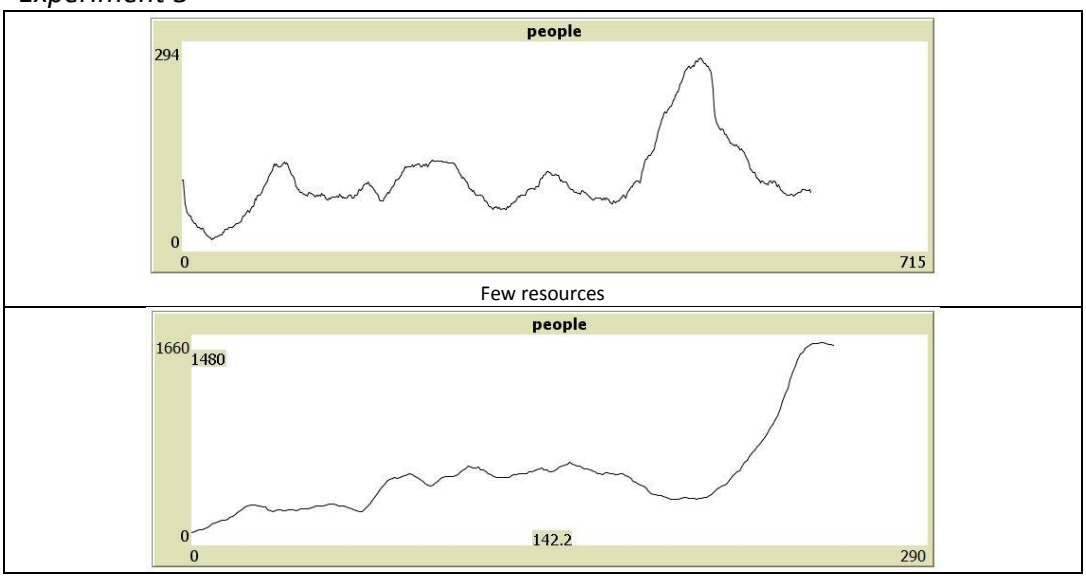
*Experiment 2* - seed 1 = 1942002165, seed 2 = -1680074612, switch = “redistribution?”



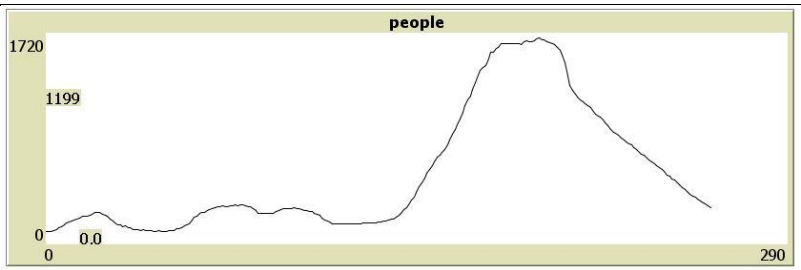



There is no evidence that redistribution influences wealth or income mean but it has a relevant expansive effect on population number and on mean age of agents.

*Experiment 3*



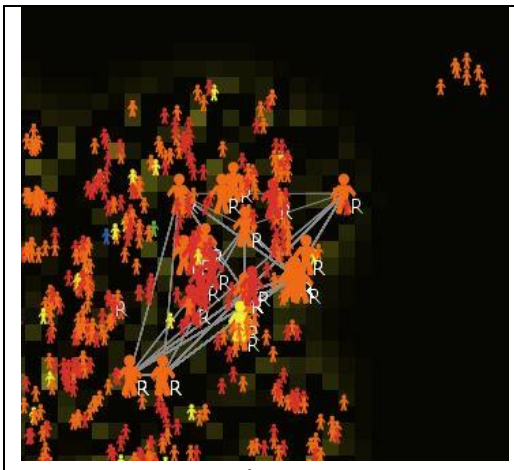
Abundant resources



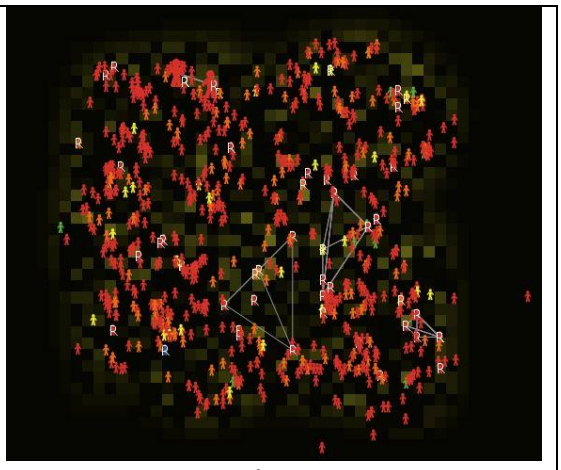

Abundant resources

Introducing redistribution around ticks = 200 can cause a sudden growth of our population but this is evident only if resources are abundant. On the other hand, removing redistribution often produces a quick decrease in the number of agents (death is present, of course).

*Experiment 4*



Tol 0.25



Tol 0.06

Representatives can have different behavior, for example they, if tolerance is quite high, it happens they move together spontaneously while, when tolerance is low, they form “independent” groups.

**Conclusion**

*Things to try*

I suggest trying different combinations of patch parameters and different dimension of the “fertile rectangle”. Furthermore the observer can choose to activate only a part of the many switches present in the model.
It could be interesting to change maximum grain available on patches, creating artificial economic crisis or expansion, or to simulate a catastrophe with button “black death”.

*Extending the model*

This model could be improved in a huge number of ways: public services and public costs might be included, different electoral rules could be implemented.
It would be challenging to introduce competition or cooperation between turtles, as well making them change preferences according to a suitable happiness indicator.

*Related models*


- Wilensky, U. (1998). NetLogo Wealth Distribution model. Center for Connected Learning and Computer-Based Modeling, Northwestern Institute on Complex Systems, Northwestern University, Evanston, IL.
- Jan C. Thiele, Boxplot example of Rserve-extension, University of Goettingen.