Consumer Choices with Bayes’ Theorem

Riccardo Bastianutti Stefano Butelli

University of Turin

April 2019

1. Introduction

The economy is structured as a social science during the positivist period, thus inheriting a strictly rational vision.
The mathematical models that have arisen in this context are the basis of the traditional economy, but compared with reality they have shown anomalies. For example, traditional economics should teach us when markets work but sometimes fails to explain why markets that should fail work and vice versa, this is due to the assumption of maximizing individual interest.

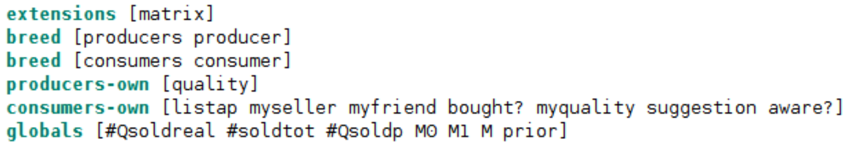
These shortcomings have given life to the branch of behavioral economics, which studies how real people make economic decisions, trying to include in the skeleton of the traditional economy a probable modeling of the vastness of decision-making processes.
For the consumer, the ultimate goal of decision-making is choice.
If we accept that consumers' economic behavior is guided by their internal states, then dialogue with behavioral and cognitive sciences is obvious.
Cognitive science studies, with a multidisciplinary theoretical and methodological approach, ourselves conceived as information processors.

We believe that, just as economic mathematical models benefit from being informed by cognitive sciences, simulations of economic contexts should also take them into account.
The agent-based simulation must therefore incorporate the discoveries of the cognitive field to give a more likely description of the irrational behaviors of the subjects involved.
In our work, we have a world divided into two types of agents: consumers and producers.

The producers will be divided into high quality and medium quality producers. Consumers will have a variable chance of recognizing the quality of products.
Here we will consider only the internal dynamics of consumers and not producers. The choices of consumer agents will be potentially influenced by two factors: marketing and the so-called "word of mouth".
Within this framework the elements of the cognitive sciences that we want to implement in the description of consumers are mainly two: the first replaces the ancient image of the perfectly rational consumer by taking advantage of the Bayesian approach which in several studies proved to be an excellent approximation of the choices of the individual; the second is taken from the article by Johnson et al. (2018) which demonstrates how the great trust in some of our peers can even lead to the self-denial of one's own interests and even direct knowledge.
The choices made by consumers will then be determined by the Bayes’ theorem using two ingredients:
a) trust in other consumers (word of mouth) or producer (marketing)
b) a statistical data based on the collective perception of purchases ("consumer association").

2. The simulation

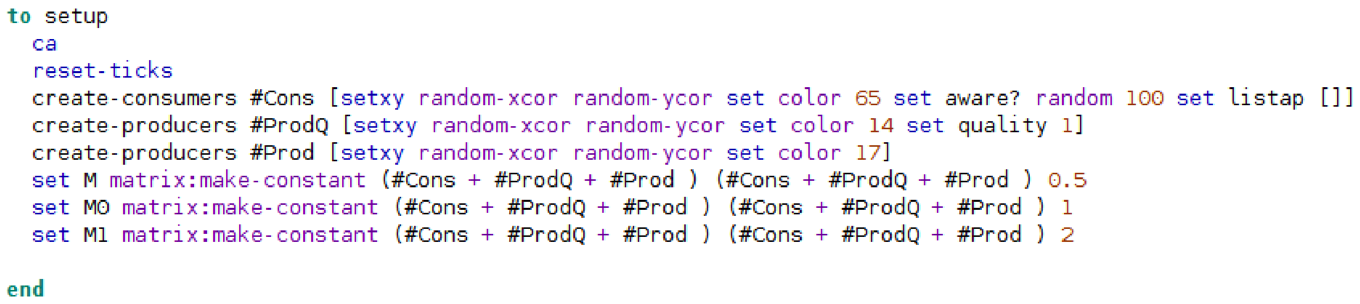
Let's now pass to a detailed comment of the code used with NetLogo 6.0.4. Within the NetLogo environment it is possible to distinguish the "turtles" (the agents) in different breeds, and here the two breeds "producers" and "consumers" are naturally created.



```
extensions [matrix]
breed [producers producer]
breed [consumers consumer]
producers-own [quality]
consumers-own [listap myseller myfriend bought? myquality suggestion aware?]
globals [#Qsoldreal #soldtot #Qsoldp M0 M1 M prior]
```

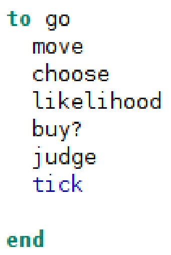
Note that the only additional variable producers have is *quality*, which determines the two possible qualities of their product. Consumers instead have a longer list of variables: *listap* contains the list of producers from which each consumer has bought during the entire run, *myseller* contains at each tick who is the potential seller, and *myfriend* which consumer acts as a "consultant", *suggestion* is a binary variable that contains the advice received (buy / not buy), *bought?* is a binary variable to determine whether or not everyone bought each tick, *myquality* saves the consumer's judgment on the purchase made in the current tick, and finally *aware?* contains the probability (in percentage) that the consumer recognizes the quality of the product purchased,and is used only if the *uniform-awareness*-switch,is-Off.
The global variables introduced will instead serve to contain: *#Qsoldreal* the number of quality products sold, *#soldtot* the number of products sold, *#Qsoldp* the number anticipated by consumers of quality products sold, *M0* will be the matrix that counts the number of successes totals for each pair of agents (for each purchase is considered successful among consumers if the advice received matches the perceived quality, while success between consumer and producer if the perceived quality is good), *M1* that counts the number of total interactions (see over for interaction), and *M* "trust matrix" the division between successes and totals; finally, *prior* is the data given at each tick by the "consumer association": the number of quality products sold perceived divided by the number of total purchases, calculated at each tick, is used by each consumer in the next tick as "bias" on the relative population of quality producers.

The setup creates the agents in the desired number and initializes the matrices as constants, initially assigning a success over two to each couple of agents and thus obtaining a constant "trust matrix" 0.5



```
to setup
  ca
  reset-ticks
  create-consumers #Cons [setxy random-xcor random-ycor set color 65 set aware? random 100 set listap []]
  create-producers #ProdQ [setxy random-xcor random-ycor set color 14 set quality 1]
  create-producers #Prod [setxy random-xcor random-ycor set color 17]
  set M matrix:make-constant (#Cons + #ProdQ + #Prod ) (#Cons + #ProdQ + #Prod ) 0.5
  set M0 matrix:make-constant (#Cons + #ProdQ + #Prod ) (#Cons + #ProdQ + #Prod ) 1
  set M1 matrix:make-constant (#Cons + #ProdQ + #Prod ) (#Cons + #ProdQ + #Prod ) 2

end
```

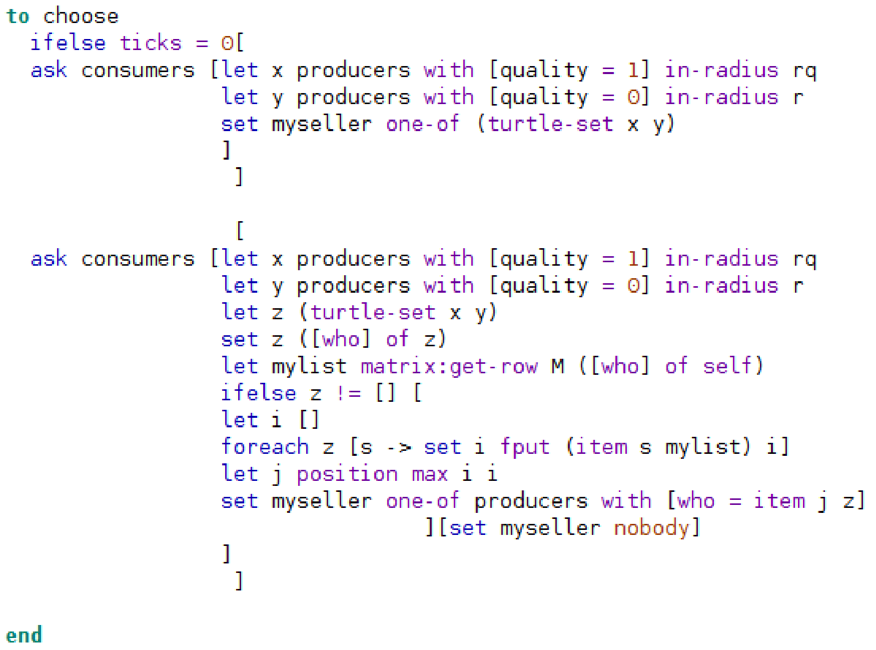
Consumers will perform the following list of actions for each tick:



```
to go
  move
  choose
  likelihood
  buy?
  judge
  tick

end
```

- move: each consumer moves one step in a random direction (here the *bought?* variable is reset to all)

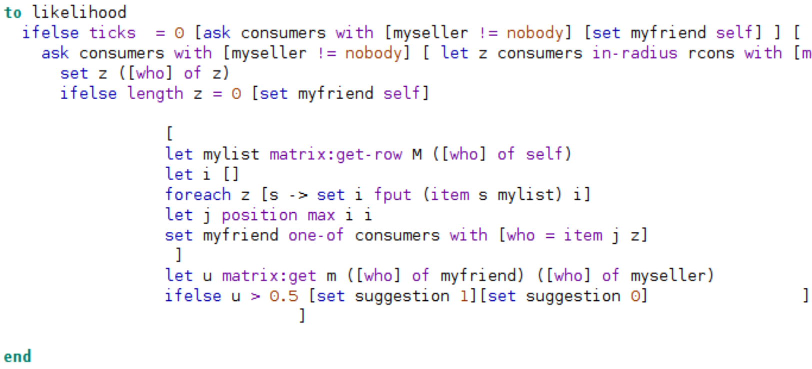
- choose: each producer has a certain range of action, so in this phase each consumer must choose her potential seller for this tick, choosing from those who are within range. In the first tick the choice is made randomly, from the second each consumer chooses between the ones she "sees" the producer she trusts most, with whom she has recorded a higher success rate (data contained in the matrix *M*). Each saves the choice in *myseller*.



```
to choose
  ifelse ticks = 0[
  ask consumers [let x producers with [quality = 1] in-radius rq
                 let y producers with [quality = 0] in-radius r
                 set myseller one-of (turtle-set x y)
                 ]
                  ]

                  [
  ask consumers [let x producers with [quality = 1] in-radius rq
                 let y producers with [quality = 0] in-radius r
                 let z (turtle-set x y)
                 set z ([who] of z)
                 let mylist matrix:get-row M ([who] of self)
                 ifelse z != [] [
                 let i []
                 foreach z [s -> set i fput (item s mylist) i]
                 let j position max i i
                 set myseller one-of producers with [who = item j z]
                                 ][set myseller nobody]
                 ]
                  ]

end
```

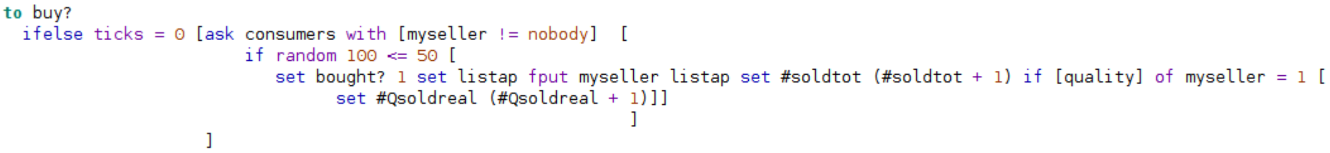
- likelihood: starting from the second tick, each consumer chooses from consumers she can "see" (in a certain radius common to all consumers) those who have purchased at least once from the producer chosen in the previous point, and among these, chooses the consumer with which she recorded a higher success rate (data contained in the matrix *M*). Depending on the story between the chosen consumer and the *myseller* producer it is decided whether the advice is to buy or not, and saved in *suggestion*.

In the event that no one is present, or no one has yet purchased from the chosen manufacturer, or at the first tick for all, the *myfriend* variable is occupied by self, meaning that each consumer proceeds with the purchase without a predominant advice (see next point) or with her own.



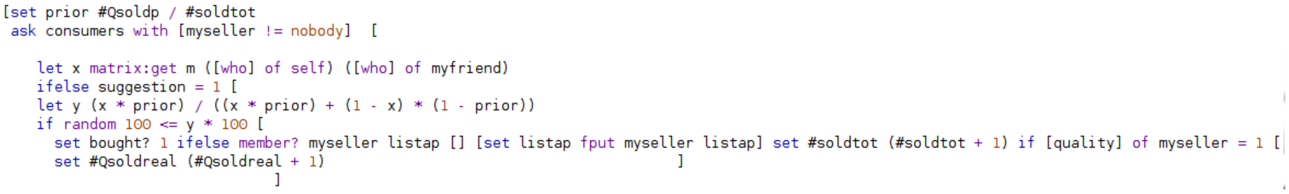
```
to likelihood
  ifelse ticks  = 0 [ask consumers with [myseller != nobody] [set myfriend self] ] [
    ask consumers with [myseller != nobody] [ let z consumers in-radius rcons with [member? [myseller] of self listap]
      set z ([who] of z)
      ifelse length z = 0 [set myfriend self]

                 [
                 let mylist matrix:get-row M ([who] of self)
                 let i []
                 foreach z [s -> set i fput (item s mylist) i]
                 let j position max i i
                 set myfriend one-of consumers with [who = item j z]
                  ]
                 let u matrix:get m ([who] of myfriend) ([who] of myseller)
                 ifelse u > 0.5 [set suggestion 1][set suggestion 0]                ]
                              ]

end
```

- buy? : each consumer decides whether or not she buys on the basis of the information received. At the initial tick when the matrices are all constant, the purchase is made with a 50% probability through the use of the random function.



```
to buy?
 ifelse ticks = 0 [ask consumers with [myseller != nobody]  [
                        if random 100 <= 50 [
                           set bought? 1 set listap fput myseller listap set #soldtot (#soldtot + 1) if [quality] of myseller = 1 [
                                 set #Qsoldreal (#Qsoldreal + 1)]]
                                                                                       ]
                      ]
```

At each subsequent tick, instead, *prior* is updated and the probability of buying is calculated by Bayes' theorem combining *prior* and reliability towards the consumer chosen as *myfriend*.



```
[set prior #Qsoldp / #soldtot
 ask consumers with [myseller != nobody]  [

    let x matrix:get m ([who] of self) ([who] of myfriend)
    ifelse suggestion = 1 [
    let y (x * prior) / ((x * prior) + (1 - x) * (1 - prior))
    if random 100 <= y * 100 [
      set bought? 1 ifelse member? myseller listap [] [set listap fput myseller listap] set #soldtot (#soldtot + 1) if [quality] of myseller = 1 [
      set #Qsoldreal (#Qsoldreal + 1)                                       ]
                           ]
```

In the event that the purchase is made the variables *bought ?*, *listap*, *#soldtot*, and *#Qsoldreal* are updated. Only the case in which the advice to buy is shown (*suggestion* = 1), the complementary case isn't shown here but can be found in the .nlogo file uploaded online and is obviously specular.

In the case in which the *marketing* switch is On, if the purchase has not been made there is a second possibility to buy with probability still calculated through the Bayes' theorem but combining *prior* with the reliability towards the producer increased by a certain percentage determined by *marketing-strength*. If the purchase is made, the correct variables are then updated.

```
if marketing [ask consumers with [myseller != nobody] with [bought? = 0] [
  let x (matrix:get m ([who] of self) ([who] of myseller) + marketing-strenght * matrix:get m ([who] of self) ([who] of myseller))
  if x >= 1 [set x 0.99]
  let y (x * prior) / ((x * prior) + (1 - x) * (1 - prior))
   if random 100 <= y * 100 [
     set bought? 1 ifelse member? myseller listap [] [set listap fput myseller listap] set #soldtot (#soldtot + 1) if [quality] of myseller = 1 [
     set #Qsoldreal (#Qsoldreal + 1)                                          ]
                              ]
```

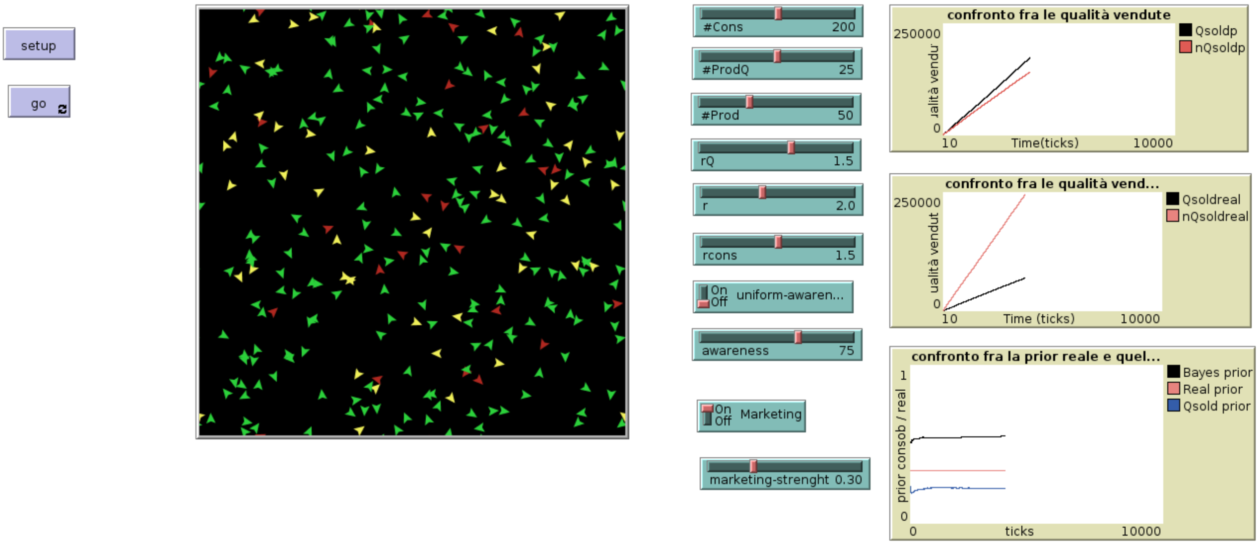
- judge: each of the consumers who has bought (ie with *bought?* = 1) has a certain probability of correctly identifying the quality of the product purchased. If the *uniform-awareness* switch is On, the global *awareness* variable is used, if it is Off, each consumer uses its own *aware?* variable. The following are the current matrix values that count the number of times each agent has been *myseller* or *myfriend* for each agent (*M1*), and the number of total successes received (*M0*).

```
to judge
  ask consumers with [bought? = 1] [ifelse uniform-awareness [
    ifelse random 100 <= awareness [set myquality [quality] of myseller] [set myquality (([quality] of myseller + 1) mod 2)]] [
    ifelse random 100 <= aware? [set myquality [quality] of myseller] [set myquality (([quality] of myseller + 1) mod 2)]]
    let x matrix:get m0 ([who] of self) ([who] of myfriend)
    let y matrix:get m1 ([who] of self) ([who] of myfriend)
    let i matrix:get m0 ([who] of self) ([who] of myseller)
    let j matrix:get m1 ([who] of self) ([who] of myseller)
```

Finally, depending on the variables *myquality* and *suggestion*, the matrix values are updated (here two of four cases are presented, the other two are similar) which will then be used on the subsequent ticks.

```
ifelse suggestion = 1 [ifelse myquality = 0 [matrix:set m1 ([who] of self) ([who] of myseller) j + 1
                       matrix:set m1 ([who] of self) ([who] of myfriend) y + 1
                       matrix:set m ([who] of self) ([who] of myseller) (i / (j + 1))
                       matrix:set m ([who] of self) ([who] of myfriend) (x / (y + 1))]
                      [matrix:set m1 ([who] of self) ([who] of myseller) j + 1
                       matrix:set m1 ([who] of self) ([who] of myfriend) y + 1
                       matrix:set m0 ([who] of self) ([who] of myseller) i + 1
                       matrix:set m0 ([who] of self) ([who] of myfriend) x + 1
                       matrix:set m ([who] of self) ([who] of myseller) ((i + 1) / (j + 1))
                       matrix:set m ([who] of self) ([who] of myfriend) ((x + 1) / (y + 1))
    set #Qsoldp (#Qsoldp + 1)]]
```

To conclude the presentation of the simulation, we show as an example the graphic interface of the project in NetLogo 6.0.4:


setup
go
#Cons 200
#ProdQ 25
#Prod 50
rQ 1.5
r 2.0
rcons 1.5
On Off uniform-awaren...
awareness 75
On Off Marketing
marketing-strenght 0.30

3. Results and Further Developments

We have carried out various simulations to check the behavior of our world as the parameters change.
At th end of this file you can check some of the simulations done so far: there are written the values of the parameters set for the given simulation and some of the results, also considered useful, and are also shown graphs at tick 10000 to see the time trend of the variables.
Being able to choose to set conscious consumers evenly or to distribute their awareness in a random way, it seemed interesting to compare the two different parameters, placing the *uniform-awareness* at the value of 50%.
From the simulations it can be seen that 50% aware consumers, with any distribution of quality and non-quality producers, believe that the world is divided equally into quality producers and poor quality producers.
On the other hand, if awareness is distributed randomly, consumers, with any distribution of producers, they see the world not distributed equally but with a deviation of about 5% in favor of higher quality producers
The first behavior is easy to analyze, in fact 50% consumers recognize the quality of the product. This implies that despite being able to find on their way only one species of producers, for them it will be possible to 50% that the producer is part of the other faction.
Instead the second behavior is more difficult to understand, but analyzing the *trust matrix* we can see that very conscious consumers trust only themselves and their fellows, while unconscious consumers have distributed trust in a random way, this implies that if in the simulation there are more producers of low quality, the conscious consumers will buy very little and therefore add little influence to the *prior*, while if the distribution of the producers were on the contrary the conscious consumers would buy more and more but balanced by unaware consumers who with their often incorrect judgment influence down the *prior*.
It can also be seen from our simulations that the turning on or off of marketing affects the total quantity of goods sold, obviously in the first case it is greater than the second, while the distribution of sales among producers is highly dependent on the range with which the producers come into contact with consumers.

Another interesting result that can be deduced from the simulations is that, if the different producers are not of the same number, increasing the awareness of consumers decreases the quantity of products sold, and this is due to the mistrust of consumers towards producers (ie the matrix confidence).

In a world populated by fully aware consumers it’s even more interesting to see that, with the passing of ticks, low quality producers would fail because they would become invisible. This is because consumers also have a trusting matrix towards producers that influences the advice consumers will give each other, or whether to buy or not. In fact, consumers always recognizing the quality of the product in the long run will advise against buying from low quality producers to other buyers. As consumer confidence increases with increasing ticks, medium quality producers will inevitably fail.
The model is still primordial so it can have several developments, we will list some of these below.
The world treated so far has a binary conception, that is quality and non-quality products, with consequent producers: we believe it can have different implications to develop a variety of qualities and relative judgment, where the consumer will have to evaluate the product based on a reference

scale. This would make the behavior of producers and consumers more real, changing the almost linear behavior.

Another step to increase the strenght of the model is to introduce the concept of price, in order to make the choice more complex as there would be two factors of influence: the quality of the product and its price.

At this point natural consequence would be to add the portion of income that can be spent by the consumer for the product in question, this would be another initial condition that would influence the choice of myseller.

Introducing the money in this model could also lead to study the strategies of producers to maximize profit, upon changing the initial conditions.

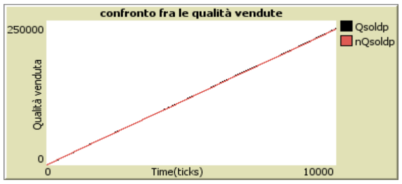
For example, producers who have more demand than supply would invest part of their profits no longer on marketing but on lowering production costs or expanding their production, so when parameters change, the best strategies for different producers could be studied.

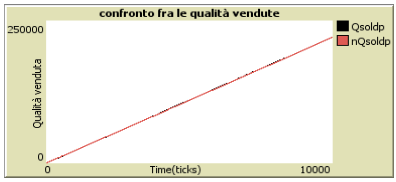
Even for consumers, by using cognitive sciences one could model the evolution of awareness due to the history of consumer interactions between producers and consumers themselves.

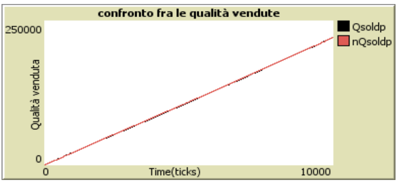
We have a lot of confidence in our model that uses Bayesian inference for decision-making, but we understand that there are several steps to be taken to achieve a realism value comparable to the complexity of the real world.

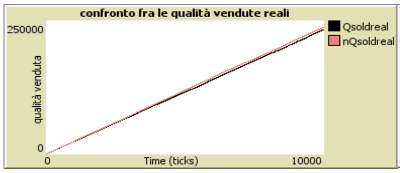
Cons = 200 ProdQ = 25 Prod = 25 rQ = 1.5 r = 1.5 rcons = 1.5 uniform on 50 Mark on 30%

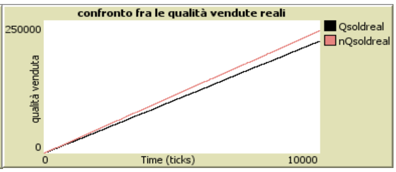
Simulazione 1	**Simulazione 2**	**Simulazione 3**
soldtot	**soldtot**	**soldtot**
471573	442910	442200
Qsoldreal	**Qsoldreal**	**Qsoldreal**
233366	211317	231750
Qsoldp	**Qsoldp**	**Qsoldp**
236184	221520	221200
nQsoldreal	**nQsoldreal**	**nQsoldreal**
238207	231593	210450
nQsolp	**nQsolp**	**nQsolp**
235389	221390	221000
prior	**prior**	**prior**
0.5008	0.500150161	0.500228428
Qsoldreal / nQsoldreal	**Qsoldreal / nQsoldreal**	**Qsoldreal / nQsoldreal**
0.9797	0.912449858	1.101
Qsoldp / nQsoldp	**Qsoldp / nQsoldp**	**Qsoldp / nQsoldp**
1.0034	1.001	1.001

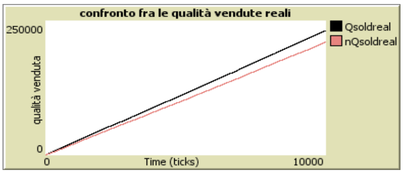


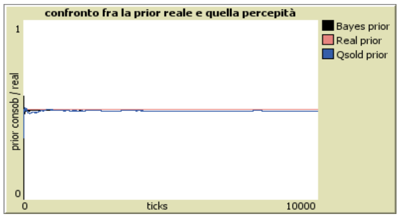


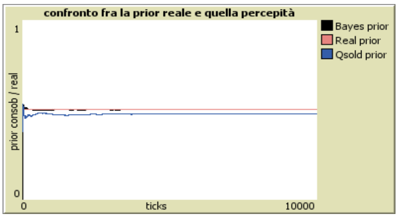


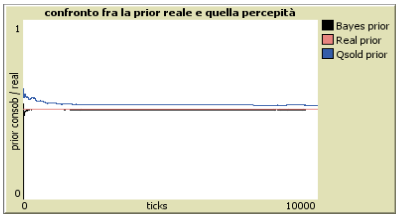






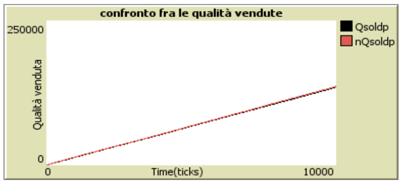


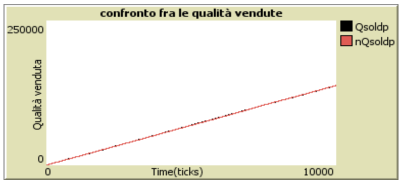


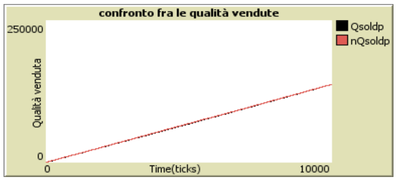


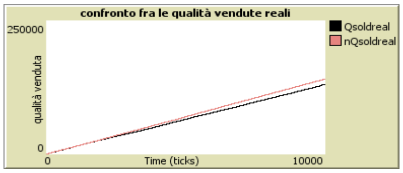
Cons = 200 ProdQ = 25 Prod = 25 rQ = 1.5 r = 1.5 rcons = 1.5 uniform on 50 Mark off

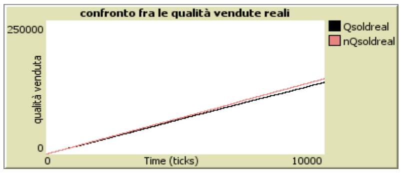
Simulazione 1	**Simulazione 2**	**Simulazione 3**
soldtot	**soldtot**	**soldtot**
271273	275585	274454
Qsoldreal	**Qsoldreal**	**Qsoldreal**
130536	134336	141028
Qsoldp	**Qsoldp**	**Qsoldp**
135062	137767	137174
nQsoldreal	**nQsoldreal**	**nQsoldreal**
140737	141249	133426
nQsolp	**nQsolp**	**nQsolp**
136211	137818	137280
prior	**prior**	**prior**
0.4979	0.499909275	0.499806878
Qsoldreal / nQsoldreal	**Qsoldreal / nQsoldreal**	**Qsoldreal / nQsoldreal**
0.9275	0.951058061	1.057
Qsoldp / nQsoldp	**Qsoldp / nQsoldp**	**Qsoldp / nQsoldp**
0.9916	0.999629947	0.999227855

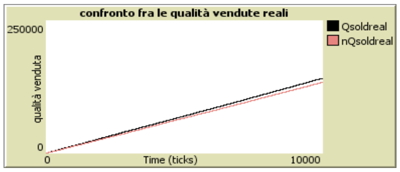


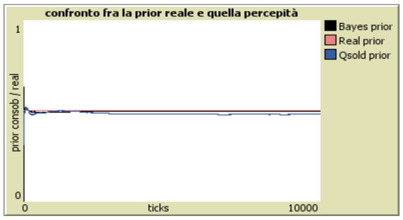


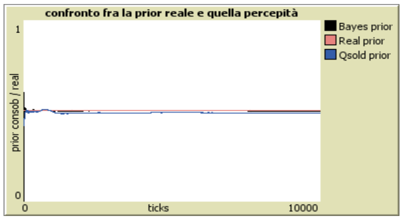


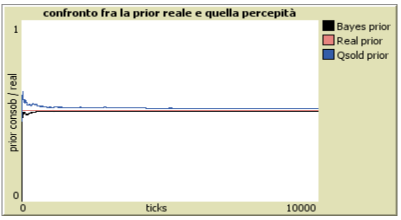






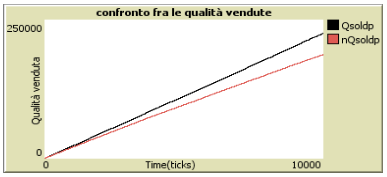


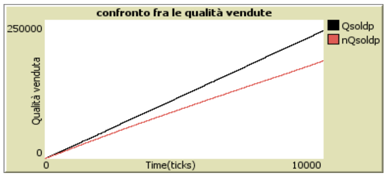


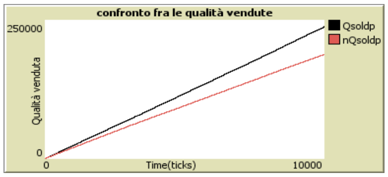


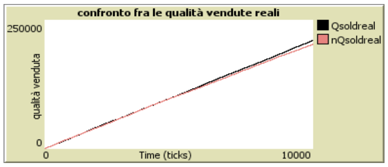
Cons = 200 ProdQ = 25 Prod = 25 rQ = 1.5 r = 1.5 rcons = 1.5 uniform off Mark on 30%

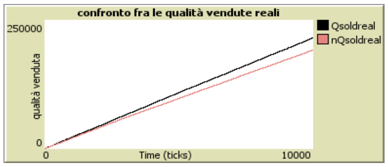
Simulazione 1	Simulazione 2	Simulazione 3
soldtot	**soldtot**	**soldtot**
412999	407411	425662
Qsoldreal	**Qsoldreal**	**Qsoldreal**
210220	215499	208569
Qsoldp	**Qsoldp**	**Qsoldp**
225750	231103	237683
nQsoldreal	**nQsoldreal**	**nQsoldreal**
202779	191912	217093
nQsolp	**nQsolp**	**nQsolp**
187249	176308	187979
prior	**prior**	**prior**
0.546605901	0.567243136	0.558387001
Qsoldreal / nQsoldreal	**Qsoldreal / nQsoldreal**	**Qsoldreal / nQsoldreal**
1.03670	1.12291	0.960735722
Qsoldp / nQsoldp	**Qsoldp / nQsoldp**	**Qsoldp / nQsoldp**
1.20561	1.31079	1.26441

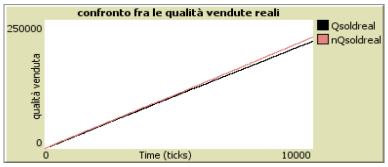


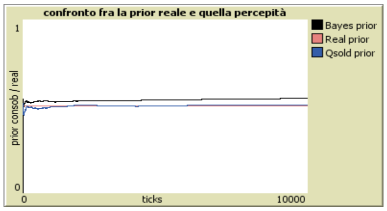


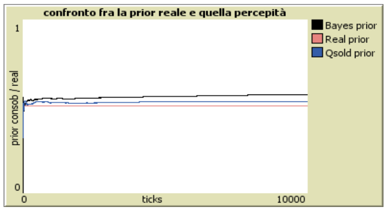


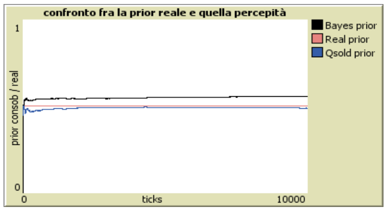






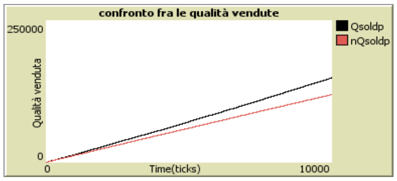


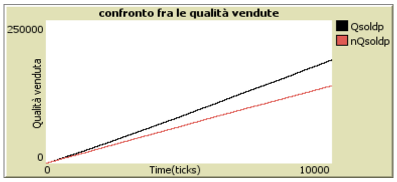


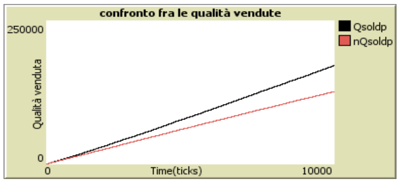


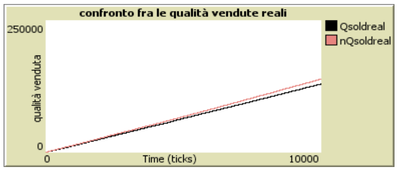
Cons = 200	ProdQ = 25	Prod = 25	rQ = 1.5	r = 1.5	rcons = 1.5	uniform off	Mark off

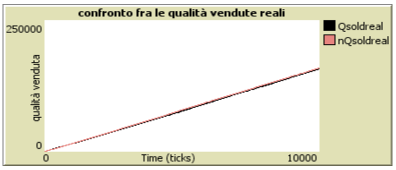
Simulazione 1	Simualzione 2	Simualzione 3
soldtot	**soldtot**	**soldtot**
267994	316571	298191
Qsoldreal	**Qsoldreal**	**Qsoldreal**
129314	157355	150268
Qsoldp	**Qsoldp**	**Qsoldp**
148722	180924	171855
nQsoldreal	**nQsoldreal**	**nQsoldreal**
138680	159216	147923
nQsolp	**nQsolp**	**nQsolp**
119272	135647	126336
prior	**prior**	**prior**
0.554953893	0.57151071	0.576333945
Qsoldreal / nQsoldreal	**Qsoldreal / nQsoldreal**	**Qsoldreal / nQsoldreal**
0.932463225	0.988311476	1.01585
Qsoldp / nQsoldp	**Qsoldp / nQsoldp**	**Qsoldp / nQsoldp**
1.24691	1.33379	1.36030

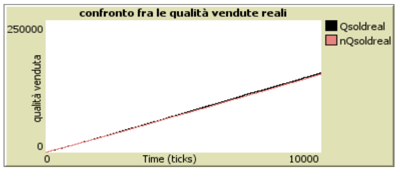


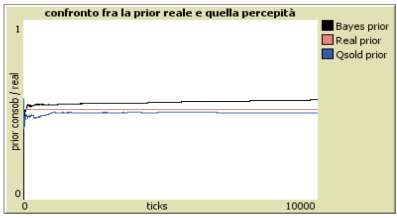


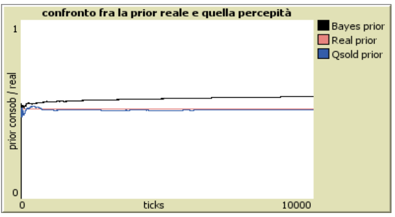


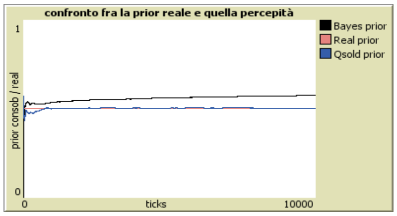






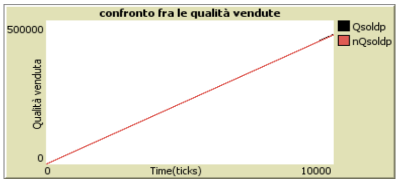


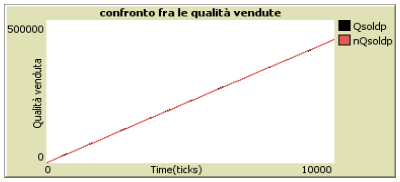


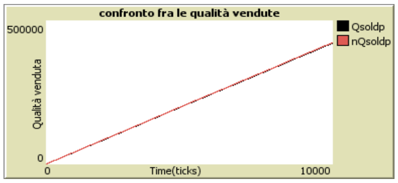


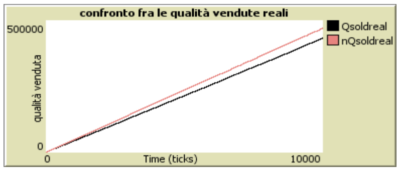
Cons = 400 ProdQ = 25 Prod = 25 rQ = 1.5 r = 1.5 rcons = 1.5 uniform on 50 Mark on 30%

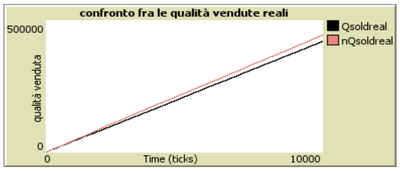
Simulazione 1	Simulazione 2	Simulazione 3
soldtot	**soldtot**	**soldtot**
903125	864980	845971
Qsoldreal	**Qsoldreal**	**Qsoldreal**
432852	420280	413838
Qsoldp	**Qsoldp**	**Qsoldp**
451685	432366	422170
nQsoldreal	**nQsoldreal**	**nQsoldreal**
470273	444700	432133
nQsolp	**nQsolp**	**nQsolp**
451440	432614	423801
prior	**prior**	**prior**
0.500134547	0.499861256	0.499044165
Qsoldreal / nQsoldreal	**Qsoldreal / nQsoldreal**	**Qsoldreal / nQsoldreal**
0.920427071	0.945086575	0.957663497
Qsoldp / nQsoldp	**Qsoldp / nQsoldp**	**Qsoldp / nQsoldp**
1.0005	0.999426741	0.996151496

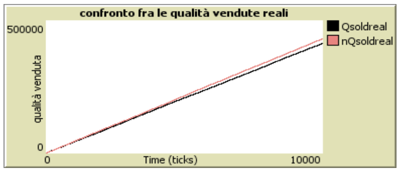


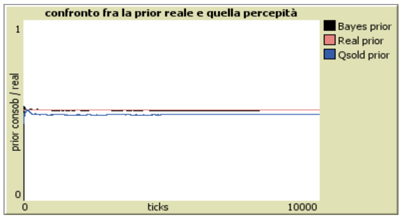


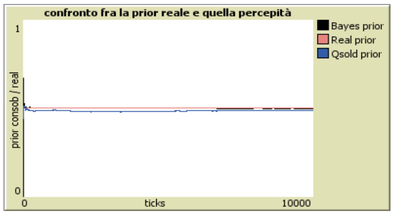


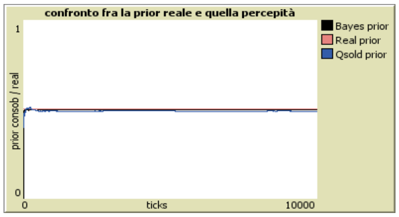






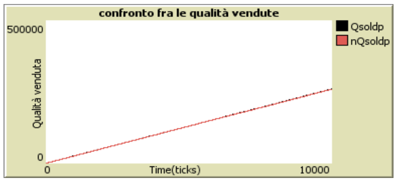


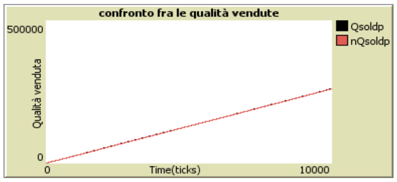


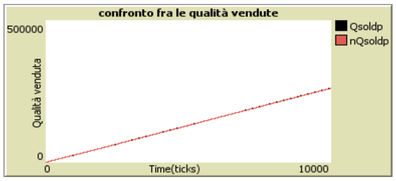


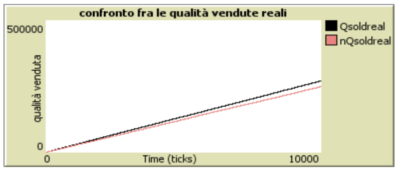
Cons = 400 ProdQ = 25 Prod = 25 rQ = 1.5 r = 1.5 rcons = 1.5 uniform on 50 Mark off

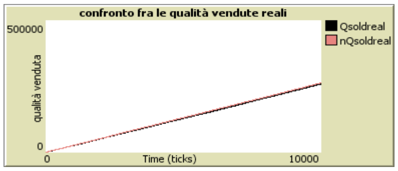
Simulazione 1	**Simulazione 2**	**Simulazione 3**
soldtot	**soldtot**	**soldtot**
519234	521670	522079
Qsoldreal	**Qsoldreal**	**Qsoldreal**
270453	258682	262543
Qsoldp	**Qsoldp**	**Qsoldp**
259936	261077	261240
nQsoldreal	**nQsoldreal**	**nQsoldreal**
248781	262988	259536
nQsolp	**nQsolp**	**nQsolp**
259298	260593	260839
prior	**prior**	**prior**
0.500620206	0.500466818	0.50039175
Qsoldreal / nQsoldreal	**Qsoldreal / nQsoldreal**	**Qsoldreal / nQsoldreal**
1.087	0.983626629	1.012
Qsoldp / nQsoldp	**Qsoldp / nQsoldp**	**Qsoldp / nQsoldp**
1.002	1.002	1.002

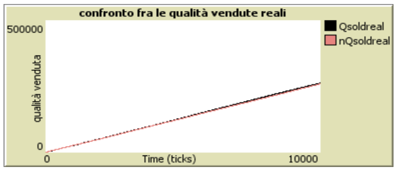


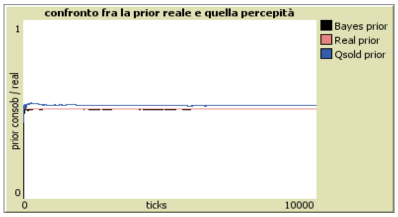


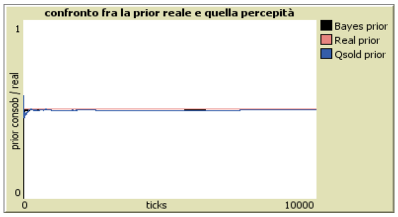


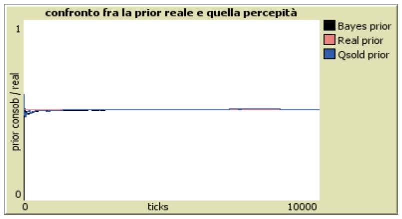












Cons = 200 ProdQ = 25 Prod = 50 rQ = 1.5 r = 1.5 rcons = 1.5 uniform on 50 Mark on 30%

Simulazione 1

soldtot

609591

Qsoldreal

214419

Qsoldp

303626

nQsoldreal

395172

nQsolp

305965

prior

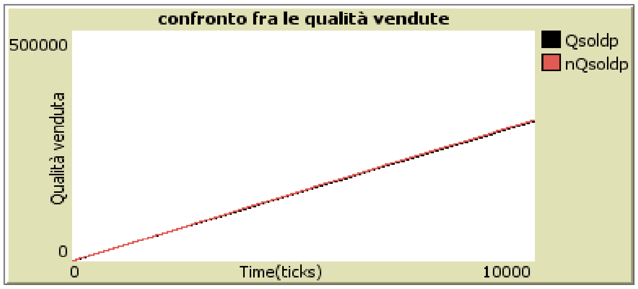
0.49807477

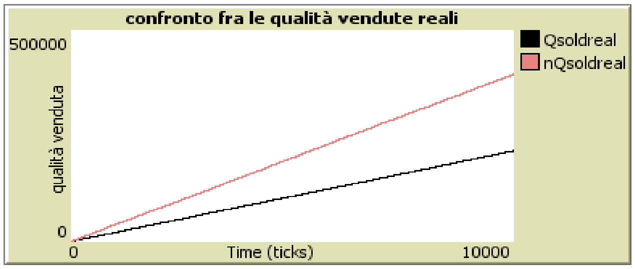
Qsoldreal / nQsoldreal

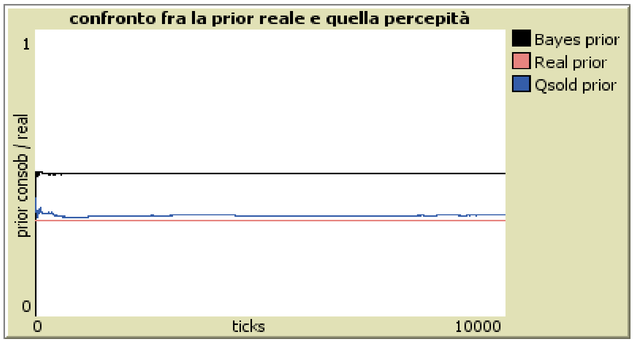
0.54259664

Qsoldp / nQsoldp

0.99235533







Cons = 200 ProdQ = 25 Prod = 25 rQ = 1.0 r = 2.0 rcons = 1.5 uniform on 50 Mark on 30%

Simulazione 1

soldtot

501438

Qsoldreal

98147

Qsoldp

247615

nQsoldreal

403291

nQsolp

253823

prior

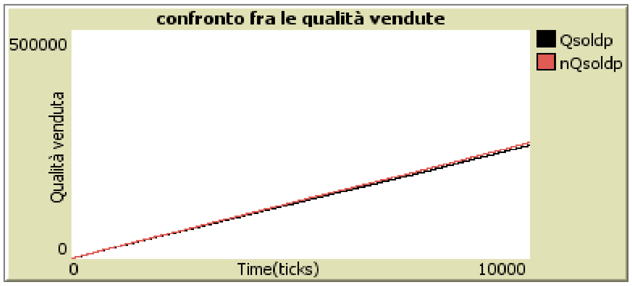
0.49381118

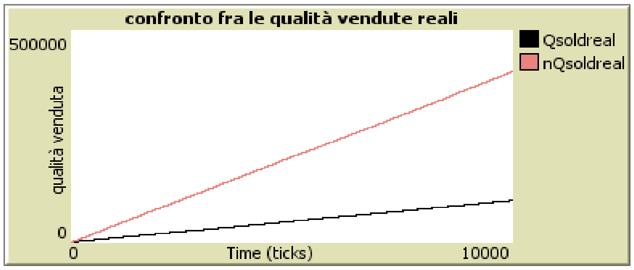
Qsoldreal / nQsoldreal

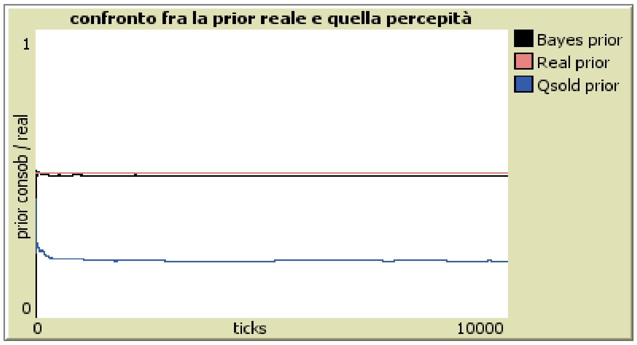
0.24336521

Qsoldp / nQsoldp

0.97554201







Cons = 200 ProdQ = 25 Prod = 25 rQ = 1.0 r = 2.0 rcons = 1.5 uniform off Mark on 30%

Simulazione 1

soldtot

486962

Qsoldreal

89811

Qsoldp

279371

nQsoldreal

397151

nQsolp

207591

prior

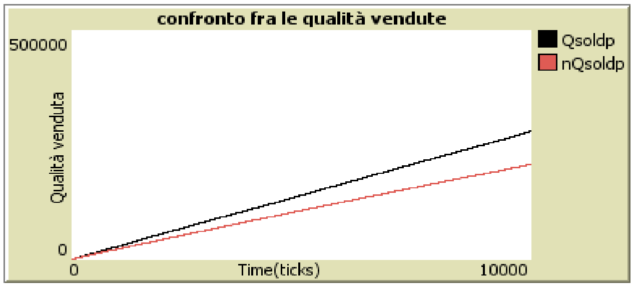
0.57370972

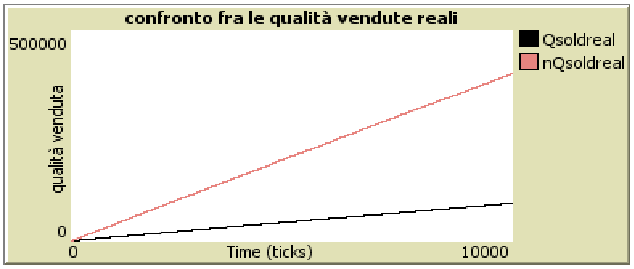
Qsoldreal / nQsoldreal

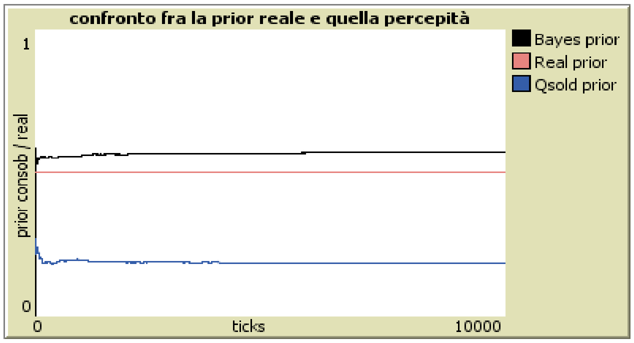
0.22613817

Qsoldp / nQsoldp

1.34577607







Cons = 200 ProdQ = 25 Prod = 25 rQ = 1.0 r = 2.0 rcons = 1.5 uniform on 50 Mark off

Simulazione 1

soldtot

303958

Qsoldreal

57526

Qsoldp

150309

nQsoldreal

246432

nQsolp

153649

prior

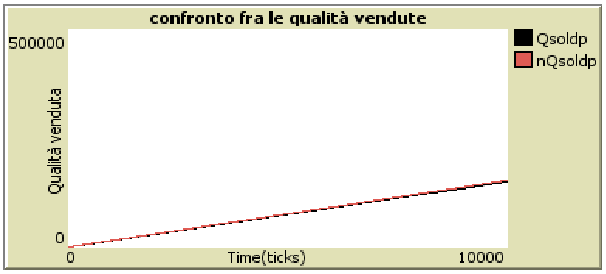
0.494509

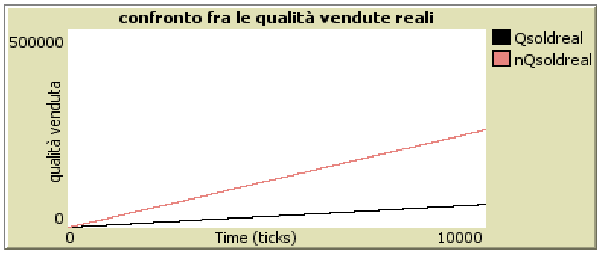
Qsoldreal / nQsoldreal

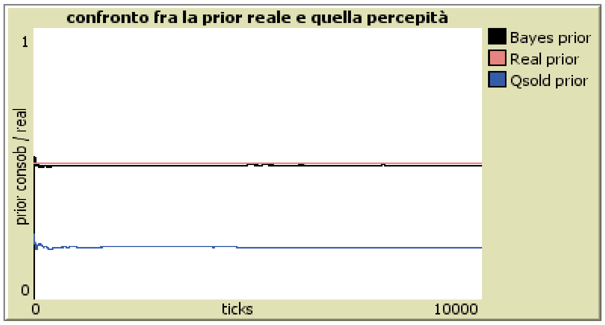
0.233436

Qsoldp / nQsoldp

0.978262







Cons = 200 ProdQ = 25 Prod = 25 rQ = 1.0 r = 2.0 rcons = 1.5 uniform off Mark off

Simulazione 1

soldtot

296800

Qsoldreal

60821

Qsoldp

163479

nQsoldreal

235979

nQsolp

133321

prior

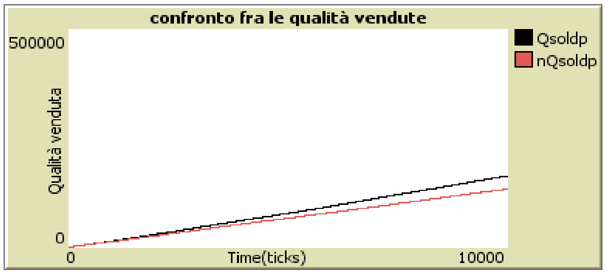
0.550799

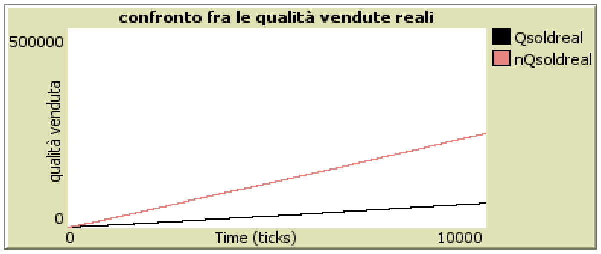
Qsoldreal / nQsoldreal

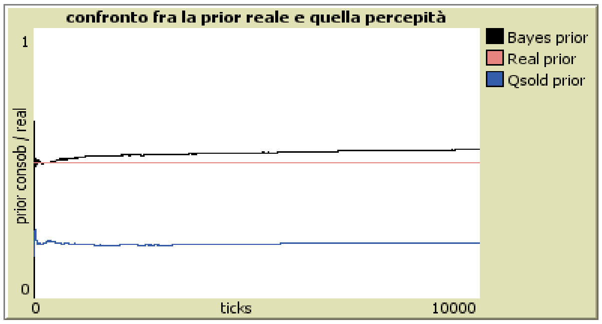
0.257739

Qsoldp / nQsoldp

1.226206

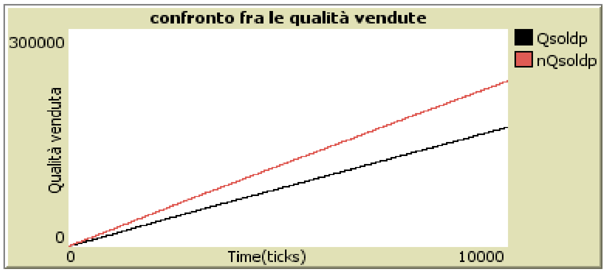


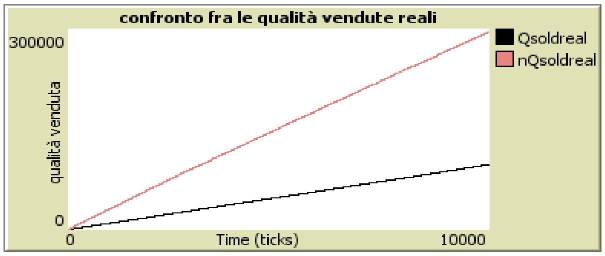


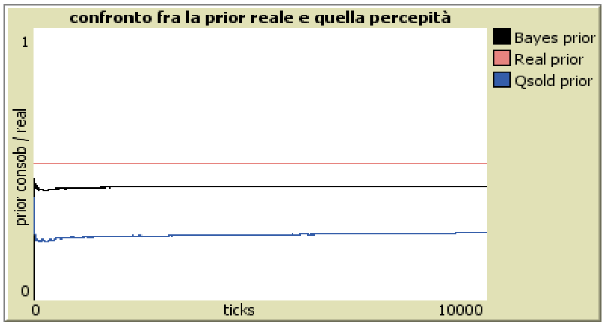


Cons = 200 ProdQ = 25 Prod = 25 rQ = 1.0 r = 2.0 rcons = 1.5 uniform on 65 Mark on 30%

Simulazione 1; 5000	**ticks 10000**
soldtot	**soldtot**
199281	393272
Qsoldreal	**Qsoldreal**
47956	97393
Qsoldp	**Qsoldp**
83124	164678
nQsoldreal	**nQsoldreal**
151325	295879
nQsolp	**nQsolp**
116157	228594
prior	**prior**
0.417119654	0.418726302
Qsoldreal / nQsoldreal	**Qsoldreal / nQsoldreal**
0.316907319	0.329164963
Qsoldp / nQsoldp	**Qsoldp / nQsoldp**
0.715617655	0.720395111

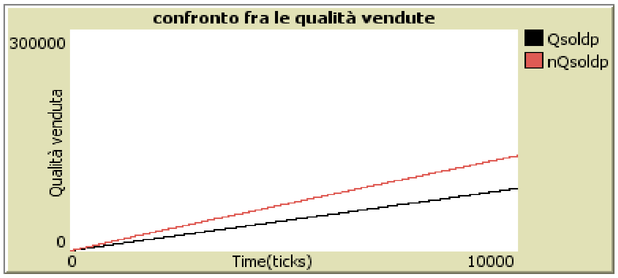


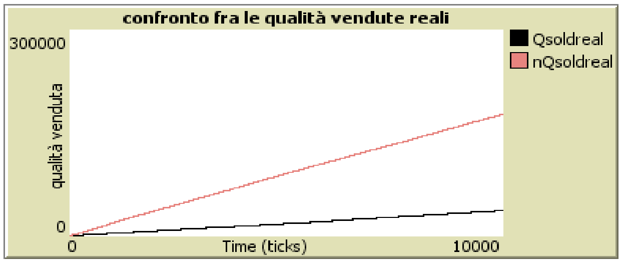


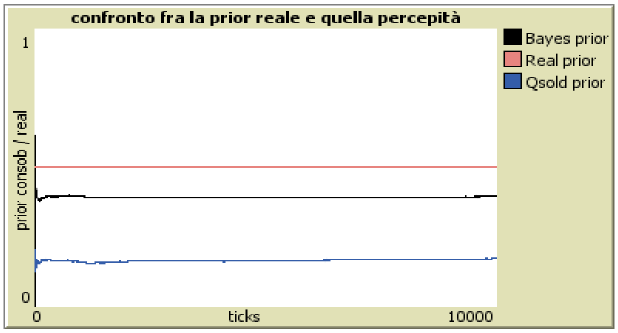


Cons = 200 ProdQ = 25 Prod = 25 rQ = 1.0 r = 2.0 rcons = 1.5 uniform on 65 Mark off

Simulazione 1; 5000	**10000 ticks**
soldtot	**soldtot**
108351	213758
Qsoldreal	**Qsoldreal**
17750	36740
Qsoldp	**Qsoldp**
42541	84628
nQsoldreal	**nQsoldreal**
90601	177018
nQsolp	**nQsolp**
65810	129130
prior	**prior**
0.392600661	0.395904197
Qsoldreal / nQsoldreal	**Qsoldreal / nQsoldreal**
0.195913952	0.207549515
Qsoldp / nQsoldp	**Qsoldp / nQsoldp**
0.646421516	0.655370557

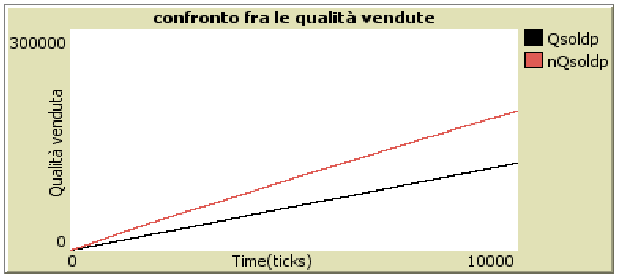


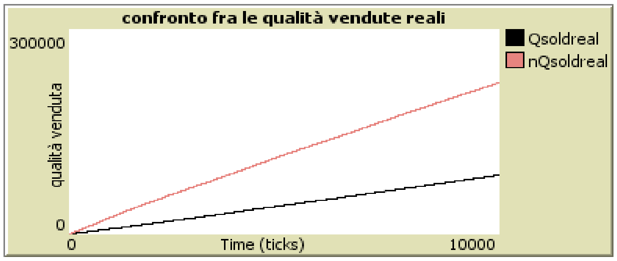


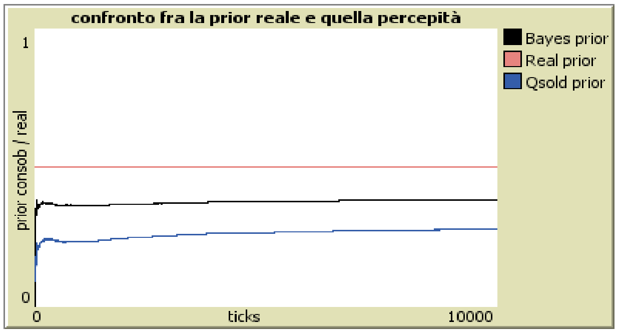


Cons = 200	ProdQ = 25	Prod = 25	rQ = 1.0	r = 2.0	rcons = 1.5	uniform on 75	Mark on 30

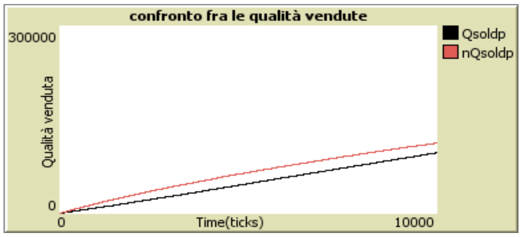
Simulazione 1; 5000	**ticks 10000**
soldtot	**soldtot**
158276	308296
Qsoldreal	**Qsoldreal**
42017	86341
Qsoldp	**Qsoldp**
59715	118738
nQsoldreal	**nQsoldreal**
116259	221955
nQsolp	**nQsolp**
98561	189558
prior	**prior**
0.377262386	0.385128932
Qsoldreal / nQsoldreal	**Qsoldreal / nQsoldreal**
0.361408579	0.389002275
Qsoldp / nQsoldp	**Qsoldp / nQsoldp**
0.605868447	0.626394032

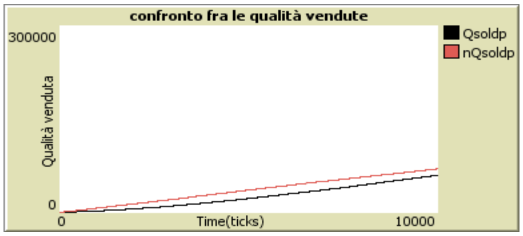


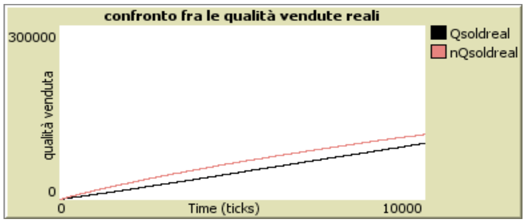


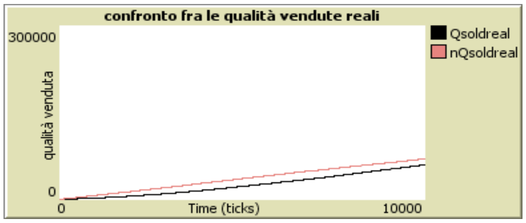


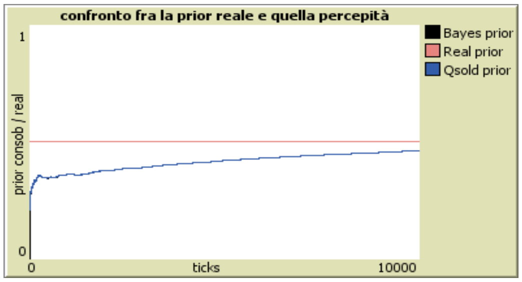
Cons = 200	ProdQ = 25	Prod = 25	rQ = 1.0	r = 2.0	rcons = 1.5	uniform on 99	Mark on 30
Simulazione 1; 5000	ticks 10000		Marketing off→		Simulazione 1;	ticks 10000	
soldtot	**soldtot**				**soldtot**	**soldtot**	
113081	209311				58438	129895	
Qsoldreal	**Qsoldreal**				**Qsoldreal**	**Qsoldreal**	
47581	96995				22410	59919	
Qsoldp	**Qsoldp**				**Qsoldp**	**Qsoldp**	
47581	96995				22410	59919	
nQsoldreal	**nQsoldreal**				**nQsoldreal**	**nQsoldreal**	
65500	112316				36028	69976	
nQsolp	**nQsolp**				**nQsolp**	**nQsolp**	
65500	112316				36028	69976	
prior	**prior**				**prior**	**prior**	
0.420784022	0.463407759				0.38346341	0.461300977	
Qsoldreal / nQsoldreal	**Qsoldreal / nQsoldreal**				**Qsoldreal / nQsoldreal**	**Qsoldreal / nQsoldreal**	
0.726427481	0.863590228				0.62201621	0.856279296	
Qsoldp / nQsoldp	**Qsoldp / nQsoldp**				**Qsoldp / nQsoldp**	**Qsoldp / nQsoldp**	
0.726427481	0.863590228				0.62201621	0.856279296	

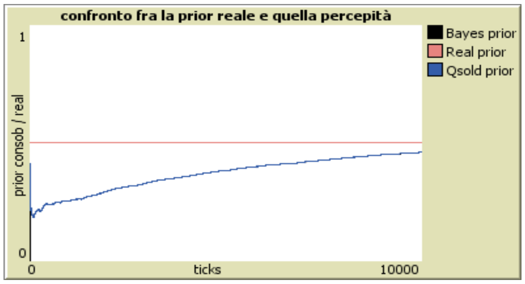

confronto fra le qualità vendute


confronto fra le qualità vendute


confronto fra le qualità vendute reali


confronto fra le qualità vendute reali


confronto fra la prior reale e quella percepità


confronto fra la prior reale e quella percepità