



**UnB**

**CIC0203 - Computação Experimental -  
TA - 2022.2 - Tarefa T6 - Aprimoramento  
de uma Simulação**

**URL Read-only Overleaf: [https:](https://www.overleaf.com/read/cytswcjsxxqh)**

**[//www.overleaf.com/read/cytswcjsxxqh](https://www.overleaf.com/read/cytswcjsxxqh)**

**Substitua a URL acima pela URL de  
visualização compartilhada de sua cópia  
inicial deste documento (share->Turno  
on link sharing -> read link, e depois  
remova esta linha de orientação.**



# Lista de tarefas pendentes



# Sumário

<b>I</b>	<b>Preparação</b>	<b>3</b>
<b>II</b>	<b>Simulação Computacional</b>	<b>5</b>
<b>1</b>	<b>T6 - Aprimoramento de uma Simulação: Laboratório e Experimento Boltzmann Wealth Model, por Fernando Ferreira Cordeiro (FernandoCordeiro)</b>	<b>7</b>
1.1	Introdução . . . . .	7
1.2	O Fenômeno do Mundo Real . . . . .	7
1.3	O Laboratório Boltzmann Wealth . . . . .	8
1.3.1	O Conceito da Simulação . . . . .	8
1.3.2	O Simulador . . . . .	9
1.3.2.1	Variáveis Independentes ou de Controle . . . . .	9
1.3.2.2	Variáveis Dependentes . . . . .	9
1.3.3	A Hipótese Causal . . . . .	10
1.3.4	O Código do Simulador . . . . .	10
1.3.4.1	Agente . . . . .	10
1.3.4.2	Modelo . . . . .	10
1.3.4.3	Servidor . . . . .	11
1.4	Os Experimentos Realizados . . . . .	12
1.5	Discussão e <i>insights</i> preliminares sobre as hipóteses . . . . .	13
1.6	Conclusão . . . . .	14
	<b>Bibliografia</b>	<b>15</b>

## *SUMÁRIO*

# Lista de Figuras

1.1	Disperção de riqueza após 100 passos de simulação em cada experimento, com num_agents = 50, prop_donation = 0.0. . . . .	12
1.2	Disperção de riqueza após 100 passos de simulação em cada experimento, com num_agents = 50, prop_donation = 0.9. . . . .	13
1.3	Disperção de riqueza após 100 passos de simulação em cada experimento, com num_agents = 50, prop_donation = 1.0. . . . .	13

# Resumo

Este documento contém o produto da tarefa especificada no título deste documento, conforme as orientações em <https://www.overleaf.com/read/cytswcjsxxqh>.





# Parte I

## Preparação



# Parte II

## Simulação Computacional



# Capítulo 1

## T6 - Aprimoramento de uma Simulação: Laboratório e Experimento Boltzmann Wealth Model, por Fernando Ferreira Cordeiro (FernandoCordeiro)

### 1.1 Introdução

Este capítulo apresenta a construção e uso do laboratório de simulações Boltzmann Wealth para a realização de experimentos que tem por objetivo investigar a hipótese causal <nomear a hipótese causal> que relaciona variáveis independentes e variáveis dependentes, supostamente presente nos estudos bibliométricos por mim realizados e disponíveis em ??.

É composto por mais cinco seções:

1. Descrição do fenômeno real;
2. Apresentação do laboratório de simulações;
3. Apresentação de análises exploratórias dos dados de experimentos realizados com o uso do laboratório;
4. Discussão sobre *insights* obtidos após os experimentos; e
5. Conclusões.

### 1.2 O Fenômeno do Mundo Real

O fenômeno modelado pelo laboratório é a transmissão de um vírus em um grupo de pessoas, sendo assim, um fenômeno social e econômico.

O Modelo de Riqueza de Boltzmann é um modelo teórico e não tem uma contraparte direta na vida real. No entanto, pode ser usado para analisar e entender certos fenômenos relacionados à distribuição de riqueza e desigualdade de renda.

Por exemplo, o modelo pode ser usado para estudar o impacto dos mecanismos de transferência de riqueza, como herança, tributação e caridade, na distribuição de riqueza em uma população ao exemplo do laboratório que será mostrando, onde foi utilizado uma variável de doação para caso o agente possua uma economia consideravelmente maior que a de seu vizinho. Também pode ser usado para analisar o efeito de políticas econômicas, como tributação progressiva, sobre a desigualdade de riqueza.

Outro exemplo da vida real é o estudo da distribuição de riqueza dentro do atual economia de mercado ([TOSCANI, 2010](#)) ou até pela influência do conhecimento na evolução da riqueza em um sistema de agentes que interagem por meio de transações binárias, como apresentado em ([PARESCI; TOSCANI, 2014](#)). No qual as simulações numéricas mostram que a presença do conhecimento tem o potencial de produzir uma classe de agentes ricos e de responder por uma proporção maior da desigualdade de riqueza, dentro de um ambiente de negociações de risco.

Em resumo, embora o Modelo de Riqueza de Boltzmann não seja uma representação direta de nenhum sistema da vida real, ele pode ser usado para estudar e entender a dinâmica da distribuição de riqueza e desigualdade de renda em vários cenários.

## 1.3 O Laboratório Boltzmann Wealth

O laboratório Boltzmann Wealth se trata da simulação de distribuição de riqueza entre um determinado número de agentes. A ideia é buscar metrificar como essa riqueza será distribuída entre agentes adjacentes, e como a possibilidade de doar uma parte maior de sua riqueza pode afetar em uma maior igualdade entre os agentes. O modelo utiliza do Coeficiente de Gini para medir a dispersão destinada a representar a desigualdade de riqueza dentro do mundo dos agentes.

### 1.3.1 O Conceito da Simulação

O código fonte da simulação foi obtido a partir do framework Mesa, que possui um grande número de pacotes disposto como uma biblioteca para o Python. Tal código base (`boltzmann_wealth_model`) se encontra exposto no GitHub pela desenvolvedora como um dos pacotes de exemplos e seu site principal onde o próprio modelo de boltzmann é usado como tutorial pode ser acessado aqui ([PROJECT MESA, 2021](#)).

Em oposição ao mundo real, a simulação trabalha com muitas poucas variáveis que possam influenciar diretamente e indiretamente na distribuição de riqueza, tendo um comportamento que tende a ser uniforme independente do número de agentes. Com a adição da possibilidade de trocar uma quantidade a mais de riqueza, através da chance de doação, a distribuição apresentada no Coeficiente de Gini acaba sendo relativamente previsível e segue de acordo

com a propabilidade de se doar parte da riqueza. Sendo assim, um número de suposições foi possível de serem feitas.

O mundo da simulação é representado em uma grade bidimensional, a qual forma um conjunto de espaços quadrados distintos, que chamaremos de células. Cada célula pode ocupar 1 ou 2 agentes, os agentes se deslocam na grade uma célula por vez, escolhendo uma célula adjacente aleatoriamente a cada passo.

Junto a isso caso um agente acabe sendo vizinho de outro, existe a possibilidade de eles trocarem riquezas, aonde parte da riqueza do agente é passada para seu vizinho. Agente com alguma riqueza são representados como um grande círculo laranja, já agentes que não possuem nenhuma riqueza ou são devedores, possuem a cor azul.

Como dito anteriormente, para simulação foi adicionado uma nova variável: a possibilidade de doação. Dado uma probabilidade de doar (entre 0 e 1), caso esse valor seja maior ou igual ao fator de bloqueio de doação, o agente dará:

$$riqueza\_do\_agente * (1 - probabilidade\_de\_doacao) \quad (1.1)$$

### 1.3.2 O Simulador

O framework Mesa apresenta uma interface gráfica interativa, onde podemos interagir com o mundo apresentado dos agentes através de parâmetros controlados por botões e sliders, que definem o comportamento da simulação.

#### 1.3.2.1 Variáveis Independentes ou de Controle

São as seguintes as variáveis Independentes ou de Controle, manipuláveis na interface gráfica do simulador:

**Number of agents** Quantidade de agentes participando da simulação

**Donation probability** Probabilidade de um agente realizar uma doação

#### 1.3.2.2 Variáveis Dependentes

São as seguintes as variáveis Dependentes, cujos valores são coletados e apresentados na interface gráfica do simulador:

**Wealth** Quantidade de riqueza por agente

**Coefficiente de Gini** Distribuição de riqueza a cada step da simulação, quanto mais perto de 0, mais distribuído é as riquezas e o contrário para quanto está mais perto de 1.



### 1.3.3 A Hipótese Causal

A hipótese do trabalho está relacionada ao demonstrar que quanto mais cultuado é o ato de de caridade em uma sociedade, ou seja, mais provável é de uma pessoa doar parte de sua riqueza para indivíduos com baixo poder econômico, mais igualitário será a distribuição de riqueza nesse grupo.

Dessa forma ocorreria uma queda na concentração de riqueza por um pequeno grupo social, em contraparte a aumento de riquezas por um grupo maior, o que poderia impacatar nos índices sociais daquela sociedade no mundo real por exemplo.

### 1.3.4 O Código do Simulador

#### 1.3.4.1 Agente

Listagem de Código 1.1: Código da criação de um agente.

```

1  import mesa
2  import random
3
4  class MoneyAgent(mesa.Agent):
5      """An agent with fixed initial wealth."""
6
7      def __init__(self, unique_id, model):
8          super().__init__(unique_id, model)
9          self.wealth = 1
10         self.donation_prop = model.prop_donation
11
12     def move(self):
13         possible_steps = self.model.grid.get_neighborhood(
14             self.pos, moore=True, include_center=False
15         )
16         new_position = self.random.choice(possible_steps)
17         self.model.grid.move_agent(self, new_position)
18
19     def give_money(self):
20         cellmates = self.model.grid.get_cell_list_contents([self.pos])
21         if len(cellmates) > 1:
22             other_agent = self.random.choice(cellmates)
23             donation_factor = random.uniform(0, 1)
24             if self.donation_prop >= donation_factor:
25                 donate_amount = self.wealth * (1 - self.donation_prop)
26                 other_agent.wealth += donate_amount
27                 self.wealth -= donate_amount
28             else:
29                 other_agent.wealth += 1
30                 self.wealth -= 1
31
32     def step(self):
33         self.move()
34         if self.wealth > 0:
35             self.give_money()

```

#### 1.3.4.2 Modelo

Listagem de Código 1.2: Código da criação do modelo.

```

1  import mesa
2  from mesa.time import RandomActivation
3  from mesa.datacollection import DataCollector
4  import statistics
5  from .agent import MoneyAgent
6
7  def compute_gini(model):
8      agent_wealths = [agent.wealth for agent in model.schedule.agents]
9      x = sorted(agent_wealths)
10     N = model.num_agents
11     B = sum(xi * (N - i) for i, xi in enumerate(x)) / (N * sum(x))
12     return 1 + (1 / N) - 2 * B

```

```

13
14 class MoneyModel(mesa.Model):
15     """A model with some number of agents."""
16
17     def __init__(self, N, width, height, donation_probability):
18         self.num_agents = N
19         self.prop_donation = donation_probability
20         self.grid = mesa.space.MultiGrid(width, height, True)
21         self.schedule = RandomActivation(self)
22         self.running = True
23
24         # Create agents
25         for i in range(self.num_agents):
26             a = MoneyAgent(i, self)
27             self.schedule.add(a)
28             # Add the agent to a random grid cell
29             x = self.random.randrange(self.grid.width)
30             y = self.random.randrange(self.grid.height)
31             self.grid.place_agent(a, (x, y))
32
33         self.datacollector = DataCollector(
34             model_reporters={"Gini": compute_gini},
35             agent_reporters={"Wealth": "wealth"}
36         )
37
38     def step(self):
39         self.datacollector.collect(self)
40         self.schedule.step()

```

### 1.3.4.3 Servidor

Listagem de Código 1.3: Código de inicialização do servidor.

```

1 import mesa
2 from mesa.visualization.UserParam import UserSettableParameter
3 from .model import MoneyModel
4
5
6 def agent_portrayal(agent):
7     portrayal = {"Shape": "circle", "Filled": "true", "r": 0.5}
8
9     if agent.wealth > 0:
10         portrayal["Color"] = "orange"
11         portrayal["Layer"] = 0
12     else:
13         portrayal["Color"] = "blue"
14         portrayal["Layer"] = 1
15         portrayal["r"] = 0.2
16     return portrayal
17
18
19 grid = mesa.visualization.CanvasGrid(agent_portrayal, 10, 10, 500, 500)
20
21 chart = mesa.visualization.ChartModule(
22     [
23         {"Label": "Gini",
24          "Color": "#0000FF"}
25     ],
26     data_collector_name="datacollector"
27 )
28
29 model_params = {
30     "N": UserSettableParameter(
31         "slider",
32         "Number of agents",
33         50,
34         2,
35         100,
36         1,
37         description="Choose how many agents to include in the model",
38     ),
39     "width": 10,
40     "height": 10,
41     "donation_probability": UserSettableParameter(
42         "slider",
43         "Donation probability",
44         0.1,
45         0,
46         1,
47         0.1,
48         description="Choose the probability that an agent will donate money",
49     ),

```

```
50 }  
51  
52 server = mesa.visualization.ModularServer(  
53     MoneyModel, [grid, chart], "Money Model", model_params  
54 )
```

## 1.4 Os Experimentos Realizados

Para a comprovação da hipótese, foi realizado experimentos iniciais onde gradualmente foram se aumentando a probabilidade de um agente realizar uma doação, dado uma quantia fixa de agentes. Como o coeficiente de Gini variou muito pouco entre uma probabilidade de 0 e 0.5, esse intervalo foi utilizado como dado de controle.

Utilizam-se os parâmetros padrões da simulação, listados a seguir:

- Number of agents: 50
- Donation probability: 0.0

Em seguida, foram realizadas 100 passos da simulação, com um total de 15 simulações executadas dado os mesmos parâmetros. Notou-se que o coeficiente de Gini tendeu a permanecer no intervalo entre 0.6 e 0.8, indicando que a desigualdade se mantém estável depois de um período de troca de riquezas.

E então alteramos a probabilidade de doação para 0.9, executando as mesma quantia de passos e de simulações, é observado como ocorre uma queda na velocidade de crescimento do coeficiente de Gini, assim como tende a se estabilizar mais tardeamente no intervalo de 0.4 e 0.6.

Por fim, utilizamos 100% de probabilidade de doação, e atingimos o que esperavamos, a dispersão de riqueza tendeu a zero.



Figura 1.1: Dispersão de riqueza após 100 passos de simulação em cada experimento, com  $\text{num\_agents} = 50$ ,  $\text{prop\_donation} = 0.0$ .

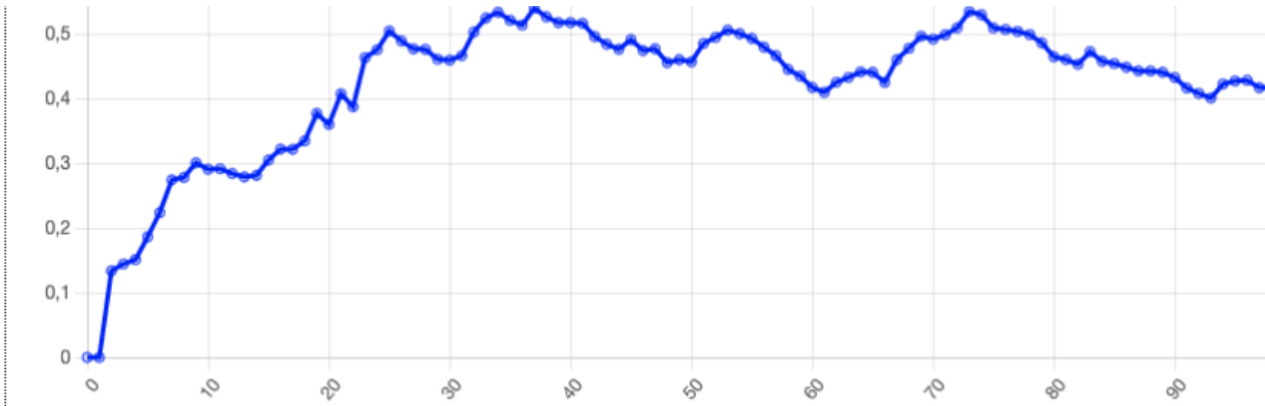


Figura 1.2: Dispersão de riqueza após 100 passos de simulação em cada experimento, com `num_agents = 50`, `prop_donation = 0.9`.

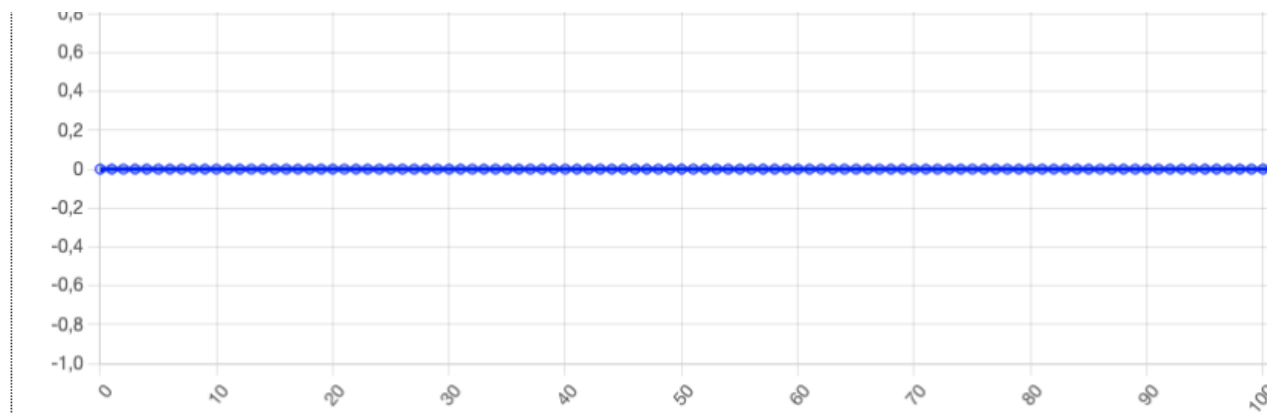


Figura 1.3: Dispersão de riqueza após 100 passos de simulação em cada experimento, com `num_agents = 50`, `prop_donation = 1.0`.

## 1.5 Discussão e *insights* preliminares sobre as hipóteses

Com os parâmetros escolhidos, a introdução do ação de doar pareceu ter efeito significativo nos experimentos, mas apenas para grandes probabilidades. Com isso, os resultados iniciais não necessariamente indicam que a hipótese está completamente correta, visto que vimos alteração significativa apenas em casos extremos. A caridade é só um dos fatores de influência, e não é esperado que seja capaz de completamente decidir a distribuição de riqueza no final da simulação.

Considerando essas observações, evidenciou-se a necessidade de criação de novas variáveis de entrada para melhor simular um ambiente real transações economicos em uma sociedade. Estudo a possibilidade de adicionar algumas novas variaveis, como a taxação de impostos de

acordo com o nível de riqueza e auxílio monetários providos dessas taxações para os agentes de melhor concentração de riqueza.

## 1.6 Conclusão

A dinâmica de experimentação desse laboratório, utilizando o framework Mesa, desmontrou ser bastante eficiente no estímulo a desenvolver uma visão crítica e experimental de fatores sociais, assim como na forma em que alteração das variáveis independentes/dependentes pode resultar em diferentes experimentos de simulação, para diferentes ambientes.

Por fim, consegue-se concluir que a hipótese causal e a simulação estão atingindo o aspecto desejado, para próxima etapa de atividades será incorporado novas variáveis que possam afetar na distribuição, assim como no refinamento do cálculo da quantidade doada. Dessa forma, espera-se trazer um ambiente com mais características que se aproximam do contexto real, como por exemplo do grau de distribuição de riquezas no Brasil.

# Bibliografia

- PARESCHI, L.; TOSCANI, G. Wealth distribution and collective knowledge: a Boltzmann approach. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, v. 372, n. 2028, p. 20130396, nov. 2014. Publisher: Royal Society. DOI: [10.1098/rsta.2013.0396](https://royalsocietypublishing.org/doi/10.1098/rsta.2013.0396). Disponível em: <<https://royalsocietypublishing.org/doi/10.1098/rsta.2013.0396>>. Acesso em: 28 jun. 2022. Citado na p. 8.
- PROJECT MESA. *Mesa: Agent-based modeling in Python 3+ — Mesa .1 documentation*. Disponível em: <<https://mesa.readthedocs.io/en/latest/>>. Acesso em: 20 jul. 2021. Citado na p. 8.
- TOSCANI, Giuseppe. Boltzmann legacy and wealth distribution. *arXiv.org, Quantitative Finance Papers*, mai. 2010. Citado na p. 8.