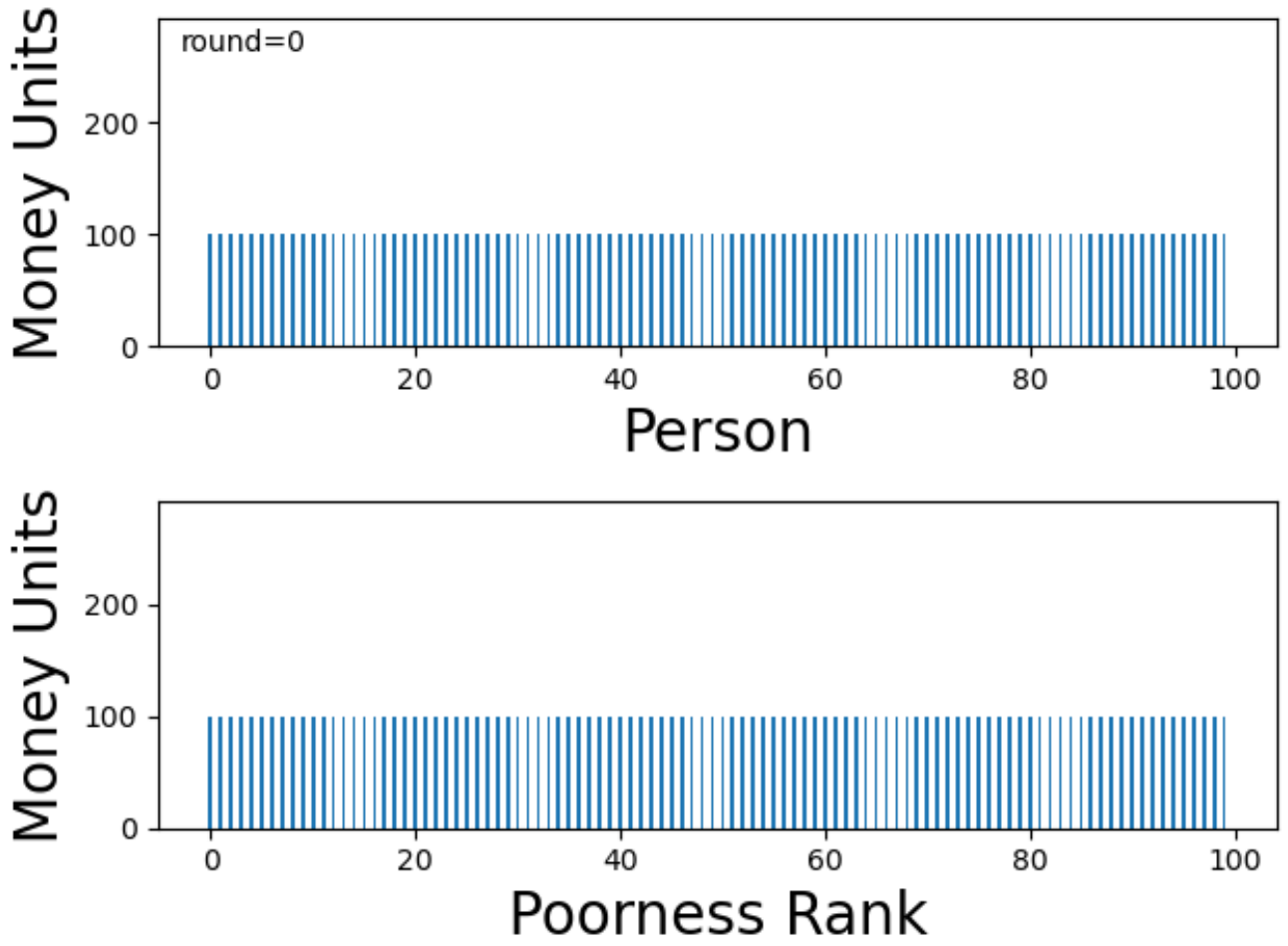

Figure 1: The initial distribution of weatlh. The person index is the x axis on top, while the "poorness rank" is on bottom. A person with poorness rank of 1 is the poorest person in this distribution.

1 Problem Statement

Suppose a group of people get together each with an initial wealth of 100 money units. We then randomly assign each person to give another person in the group 1 unit of money. This means multiple people can end up giving to the same person, and so someone can randomly accumulate wealth by being chosen (they are quite lucky). If a person has 0 units of wealth, then obviously they cannot give anyone else a unit of wealth. What sort of distribution do we expect for the wealth after many rounds of this type of interaction?

2 Answer

I am unaware of an analytic answer, though it is probably possible. In any case, we can perform simulations to arrive at the answer, and it is rather fascinating. Remember that the intial distribution looks like Figure 1.

You might think that with random assignment of money we would end up with a very even looking

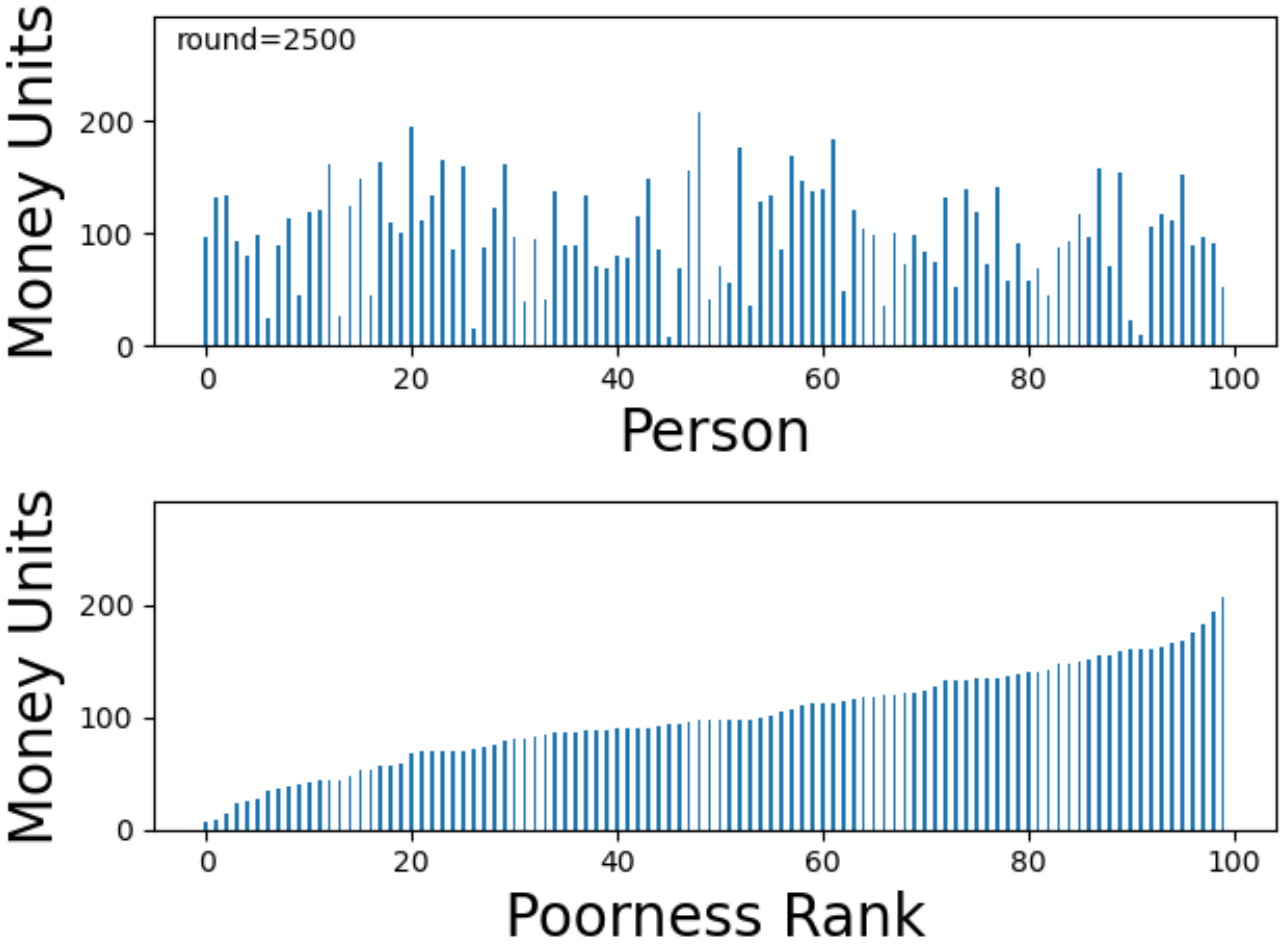

Figure 2: This shows the distribution of money after 2500 rounds of random donations. We see that the wealth distribution is extremely skewed now. In fact, it is quite far from the nice uniform distribution we began with.

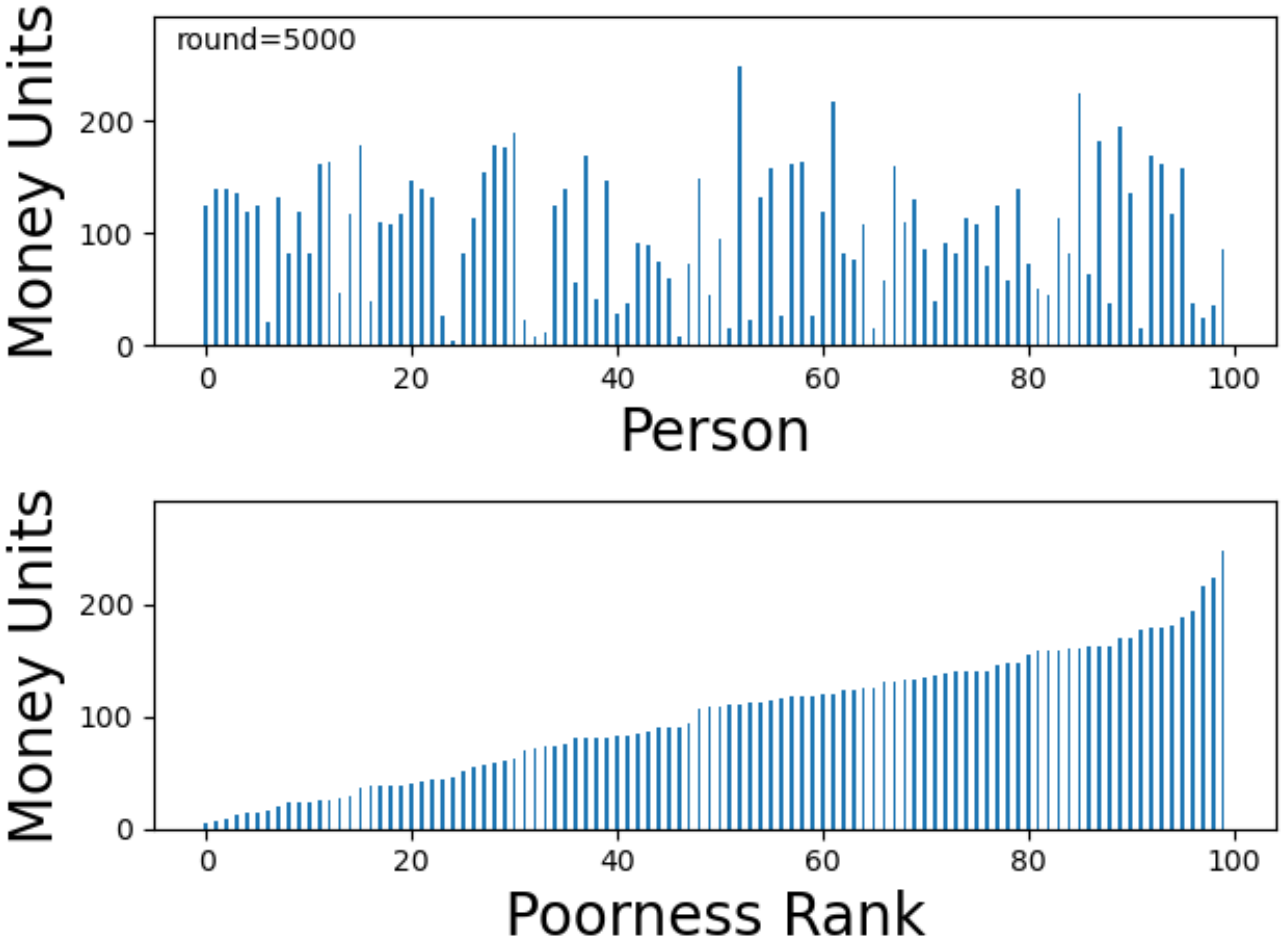

Figure 3: This shows the distribution of money after 5000 rounds of random donations. We see that the wealth distribution is even more skewed than after 2500 rounds. However, we see that the people who were wealthy in round 2500 are not necessarily the wealthy ones in round 5000. This is a key indicator of the randomness. Any single person will see their wealth vary widely from very poor to very rich.

money distribution. Your intuition is incorrect, though, as inequalities spontaneously form. What this shows is that bias is not necessary for their to be an inequality in distribution.[1] Look at Figure 2 after 2500 rounds. The distribution is not even close to even. And after 5000 rounds in Figure 3 we see that the distribution is, if anything, worse. However, there is a clear sign that there is no bias in this distribution because if you look at a single person (the person index), being wealthy at round 2500 does not necessarily mean you are wealthy at round 5000. That is, being wealthy is purely due to luck, and so any single person goes through extremes of poverty and wealth in the long run.

This is a very simplistic model and not meant to represent actual distributions of wealth in the real world. Please do not tell me it is arguing for or against anything but the statement, "Inequality always implies bias". This model disproves that statement with a counterexample. But if we changed the statement to "Inequality strongly implies bias" this example tells us nothing. That requires looking at actually realistic wealth transfer models.

The code for this is below. You can also use it to make an `mp4` animation rather than pictures at a single round if you desire. I should also have the movie in the same directory as this `pdf`.

random_distribute.py

```
#!/usr/bin/env python3

import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as anim

# initialize players and wealth
numplayers=100
rounds=5001
initwealth=100

# bank is an array with each row being a round and each column
# corresponding to a specific player
bank=np.zeros((rounds,numplayers))
# set up the initial row with everyone having the same amount
# of wealth
bank[0,:]=initwealth

# set random seed
np.random.seed(1)

for i in range(1,rounds):
# create a list of recipients for each player
# i.e., every player is assigned to give 1 unit to some other
# player
  recipients=np.random.randint(1,numplayers+1,numplayers)
  j=0
# check that a player is giving a unit to some other player
# but not to themselves.  Also check that they have money
# to spend.  If no money, assign them to player -1 which doesn't
# exist.
  while j < (numplayers):
    if bank[i-1,j]==0:
      recipients[j]=-1
      j=j+1
    elif recipients[j]==j+1:
      recipients[j]=np.random.randint(1,numplayers+1)
    else:
      j=j+1
# total units given to each player
```

[1] I must stress that that does not mean an inequal distribution is not caused by bias! Bias will certainly create an inequality. One must always look at how a distribution is created before determining if bias is a cause or random chance.

```
41    table_give=np.zeros(numplayers)
42    for j in range(numplayers):
43      table_give[j]=np.count_nonzero(recipients==j+1)
44  # first take a unit of money from everyone who has money
45    bank[i,:]=np.where(bank[i-1,:]>0,bank[i-1,:]-1,bank[i-1,:])
46  # then add units received
47    for j in range(numplayers):
48      bank[i,j]=bank[i,j]+table_give[j]
49
50  #print(bank)
51  #np.savetxt("bank.txt",bank,fmt="%d")
52
53
54  # Now animate the distribution with a bar plot.
55  width=0.35
56  xyfontsize=20
57
58  fig,axes=plt.subplots(2,1)
59  ax,ax2=axes
60  rects=ax.bar(range(numplayers),bank[0,:],width)
61  rects2=ax2.bar(range(numplayers),np.sort(bank[0,:]),width)
62
63  bankmax=1.1*np.amax(bank)
64
65  ax.set_ylim([0,bankmax])
66  ax.set_ylabel('Money Units',fontsize=xyfontsize)
67  ax.set_xlabel('Person',fontsize=xyfontsize)
68
69  ax2.set_ylim([0,bankmax])
70  ax2.set_ylabel('Money Units',fontsize=xyfontsize)
71  ax2.set_xlabel('Poorness Rank',fontsize=xyfontsize)
72  time_text=ax.text(0.02,0.95,'',transform=ax.transAxes)
73
74  ## animate by updating bar heights at each frame
75  #def animate(i):
76  #  for j in range(len(rects)):
77  #    time_text.set_text('round=%d' % i)
78  #    bank2=np.sort(bank[i,:])
79  #    rects[j].set_height(bank[i,j])
80  #    rects2[j].set_height(bank2[j])
81  #
82  #ani= anim.FuncAnimation(fig,animate,frames=rounds,blit=False,interval=1)
83  #
84  ## save
85  #ani.save('random_distribute.mp4',writer=anim.FFMpegWriter(fps=84))
86  plt.clf()
87
88  fig,axes=plt.subplots(2,1)
89  ax,ax2=axes
90  bankmax=1.1*np.amax(bank)
91
92  ax.set_ylim([0,bankmax])
93  ax.set_ylabel('Money Units',fontsize=xyfontsize)
94  ax.set_xlabel('Person',fontsize=xyfontsize)
95
96  ax2.set_ylim([0,bankmax])
97  ax2.set_ylabel('Money Units',fontsize=xyfontsize)
98  ax2.set_xlabel('Poorness Rank',fontsize=xyfontsize)
99  time_text=ax.text(0.02,0.90,'round=2500',transform=ax.transAxes)
100 ax.bar(range(numplayers),bank[2500,:],width)
101 ax2.bar(range(numplayers),np.sort(bank[2500,:]),width)
102 plt.tight_layout()
103 plt.savefig("money_a_2500.png")
104 plt.clf()
105
106 fig,axes=plt.subplots(2,1)
107 ax,ax2=axes
108 bankmax=1.1*np.amax(bank)
109
110 ax.set_ylim([0,bankmax])
111 ax.set_ylabel('Money Units',fontsize=xyfontsize)
```

```
112 ax.set_xlabel('Person',fontsize=xyfontsize)
113
114 ax2.set_ylim([0,bankmax])
115 ax2.set_ylabel('Money Units',fontsize=xyfontsize)
116 ax2.set_xlabel('Poorness Rank',fontsize=xyfontsize)
117 time_text=ax.text(0.02,0.90,'round=5000',transform=ax.transAxes)
118 ax.bar(range(numplayers),bank[5000,:],width)
119 ax2.bar(range(numplayers),np.sort(bank[5000,:]),width)
120 plt.tight_layout()
121 plt.savefig("money_a_5000.png")
122 plt.clf()
123
124 fig,axes=plt.subplots(2,1)
125 ax,ax2=axes
126 bankmax=1.1*np.amax(bank)
127
128 ax.set_ylim([0,bankmax])
129 ax.set_ylabel('Money Units',fontsize=xyfontsize)
130 ax.set_xlabel('Person',fontsize=xyfontsize)
131
132 ax2.set_ylim([0,bankmax])
133 ax2.set_ylabel('Money Units',fontsize=xyfontsize)
134 ax2.set_xlabel('Poorness Rank',fontsize=xyfontsize)
135 time_text=ax.text(0.02,0.90,'round=0',transform=ax.transAxes)
136 ax.bar(range(numplayers),bank[0,:],width)
137 ax2.bar(range(numplayers),np.sort(bank[0,:]),width)
138 plt.tight_layout()
139 plt.savefig("money_a_0000.png")
140
141
142 #plt.show()
```