Previously discussed

- Constructing network based on edges between two groups instead of probability
- Making popularity parameter based on Barabasi

What I did this week

- Reading survey literature (Book of Dickson and book of Bianconi)
- Implemented popularity parameter
- Optimizing code so it runs faster

Small analysis on total network

	edges	nodes	connected_nodes	avg indegree	max in degree	avg out degree	max out degree	reciprocity
huishouden	283790	86100	66134	4.291136	57	4.291136	57	1.000000
familie	274634	86100	72231	3.802163	57	3.802163	57	1.000000
buren	875310	86100	85903	10.189516	68	10.189516	68	0.017464
werkschool	1174412	86100	80119	14.658346	58	14.658346	58	0.000715

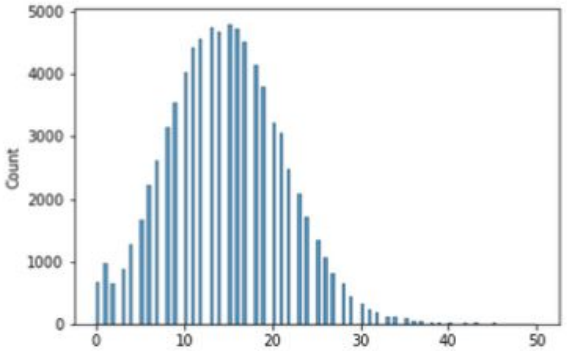
Not really representative

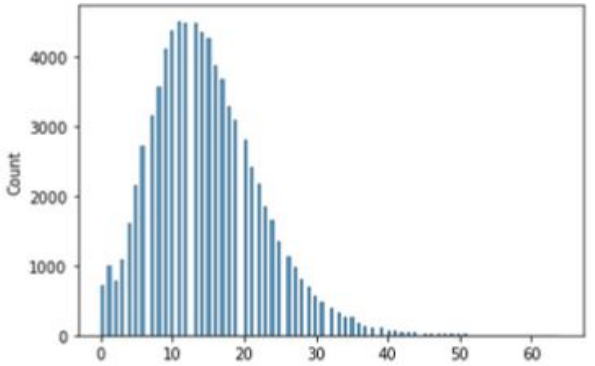
(got different values with whole network which corresponds to the data)

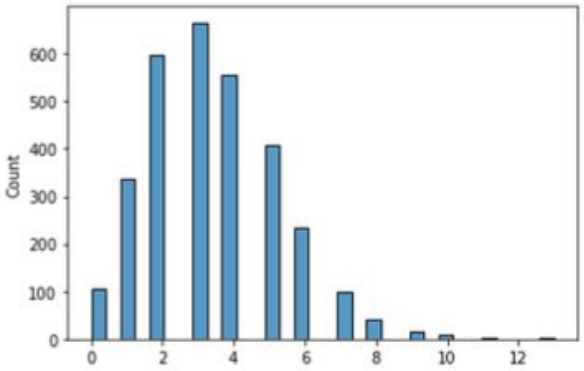
Implementation of popularity parameter

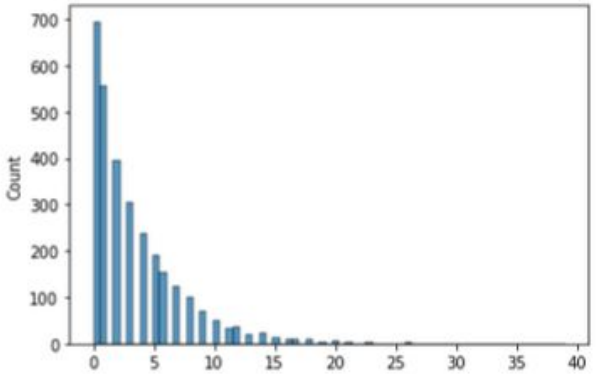
- Implemented a weight
- Each time a node gets chosen the weight gets increased by 1
- Also made a power function out of it so we can choose from a popularity parameter 0 to 1

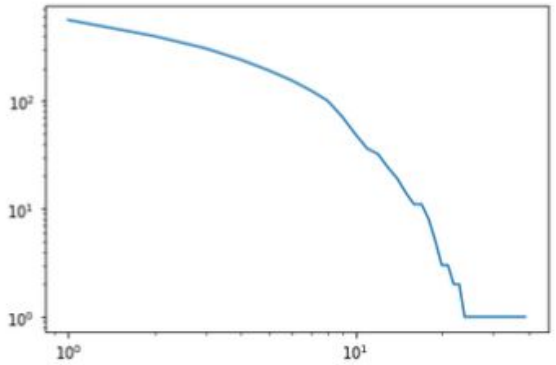
Distributions










10^2
10^1
10^0
10^0
10^1

Optimizing code

- I hashed all strings
- Used as much low level packages as possible (such as numpy)
- Discarded pandas so I do not have to iterate over these rows
- Discarded classes

- I can now run the whole network in a minute, but only without popularity parameter

Problem

- Distribution between groups looks like a exponential distribution and not a power distribution. Can maybe be fixed by increasing the popularity (increase probability of being chosen)

- I can now run the full network but not with popularity parameter (makes the programme slow)

To do

- Try to improve popularity parameter so the code can run full network
- Looking at a reciprocity parameter (now I have full reciprocity for family and household but maybe give a certain probability instead)
- Looking at spatial data and making the network based on that
- Reading more literature (finishing both books can be helpful)
- Make mono layers into a multiplex/multilayer network
- Starting already a little bit on writing the thesis so I keep structure
- Dunbar number

X -> Y then Y -> X 100%