1. A given agent in this model is initialized with two sets to represent the known/assumed group preferences and group weights of the same agent, and a pair of values to describe self weight and preference, generated via in one of 4 pseudo-random generators:
    a. Uniform: $w_i, p_i \epsilon\ u(0,1)$
    b. Discrete(A): $w_i, p_i \epsilon\ A,\ \forall a\ \in [0,1]$
    c. FoldedGaussian(μ): $c = \max(W)\,, cw_i, cp_i\ \epsilon\ |N(\mu,1)|$
    d. Beta(α, β): $w_i, p_i\ \epsilon\ \beta(\alpha, \beta)$

2. Agents calculate their decisions given the formula below, where *S, w_a and p_a* represent the set of all agents in the group, and the sets of group weights and preferences for a given agent *a* respectively:

$$d_a: \sum_{i \in S} w_{ai} p_{ai} > 0.5$$

3. Agents broadcast the decision made per the formula above to the rest of the group, which triggers all receiving agents in the group to update their preference sets notated p_a.

```
def receive(decision, sender):

if decision is True:
        preferences.set(message.sender, 1)
else preferences.set(message.sender, 0)
```

- A constant group size of 4, and calculations run in the same order per named agent for all simulations aim to investigate the scenario given in Harvey's 1974 article as closely as possible.
- Each run of the program simulates a number of groups provided in the application config.
- Program does not require any input.
- Output is in Hadoop partitioned JSON format.