# Transition Rate

Having established the equilibrium mechanics, we will now turn our attention to the transition rates. There are two rates that we must fix: the rate of transition from inactive forager to active forager, and from active forager to inactive forager.

Based on these two rates, we can establish the rate of change in the probability that our system will be in a given state at a given point. The rate of change in the probability of the system being in state at time is given by

We have already determined that the probability of a forager being active at any given time is 71.43% and the probability of them being inactive is 28.57%.

The transition rate is a matter of some interest, and the reason is the following: we have already established that this will not be a constant. The two items of particular interest are the following: first, the distribution of ant interactions is not uniform. Secondly, the rate at which harvester ants leave the nest is proportional to the number of ants that returned successfully.

Gordon *et. al.* suggested that the number of departures in the *n*th time slot can be described as a Poisson random variable of mean , where

with . is the number departing at time , and is the number arriving at time . is the rate of outgoing foragers, which increases by when a successful forager returns. decreases by for each outgoing forager. decays by during each time slot. They set based on field observations (0.15 to 1.2 ants per second) and a ants per second. They set for their simulation but indicated that other studies suggest that this parameter may be important. They swept between 0.01 and 0.25 for *c*.

Prabhakar, B., Dektar, K. N., & Gordon, D. M. (2012). The Regulation of Ant Colony Foraging Activity without Spatial Information. *PLoS Computational Biology*, *8*(8). https://doi.org/10.1371/journal.pcbi.1002670

The supplemental materials contains Matlab code that implements the above equation for several fixed values of . The values are the following:

| **Level of An** | **N** | **Dn** | |
| --- | --- | --- | --- |
| **Mean** | **Std Dev** |
| **4** | 15 | 15.6666667 | 4.5773771 |
| **5** | 15 | 16.1333333 | 4.1034248 |
| **6** | 15 | 21.2666667 | 5.5737480 |
| **7** | 15 | 26.8000000 | 9.8720384 |
| **8** | 15 | 27.9333333 | 7.3627117 |
| **9** | 15 | 29.8666667 | 9.2339952 |
| **10** | 15 | 33.8000000 | 11.1046966 |

Levene’s Test is as follows:

| **Levene's Test for Homogeneity of Dn Variance ANOVA of Squared Deviations from Group Means** | | | | |
| --- | --- | --- | --- | --- |
| **Sum of Squares** | **Mean Square** | **F Value** | **Pr > F** |
| 134543 | 22423.8 | 2.88 | 0.0126 |
| 763527 | 7791.1 |  |  |

Welch’s ANOVA is as follows

| **Welch's ANOVA for Dn** | | | |
| --- | --- | --- | --- |
| **DF** | **F Value** | **Pr > F** |
| 6.0000 | 14.26 | <.0001 |
| 42.9757 |  |  |

The distribution of is as follows:



Perhaps unsurprisingly, then, in the model, is, in fact, having an effect on (which is what we would expect, given that the model was explicitly designed for that purpose). Perhaps more relevantly for our purposes, this gives us a good sense of the relationship between and . This brings us to another issue: the means for are significantly higher than field observations suggest. Fortunately, we now have the confidence to realize that we can target our to the observed . Less fortunately, this exposes a weakness in our Matlab script: the **poissrnd** method will only return natural numbers (because it represents the number of times that the event in question happens; an event could happen, for example, 0 or 2 or 3 times in a particular time period, but it makes little sense to say that it happened 0.3 times). We could solve this in one of several ways:

* Only accept parameters that result in (since
* “Cheat” and accept (since
* While holding the other parameters equal, run the program several times where (i.e. run the simulation several times and select the results where is 1, 2, or 3, subject to the constraint that the average of all of the results for fall between 0.15 and 1.3 (inclusive) when we stop.

Our Matlab script follows the first result: only accept results where . (Note again that because ).

Obviously, there is a degree of randomness to this computation. Here are a few examples of average number of arriving ants for various values of *c* (computed using Excel’s subtotal/average function):

|  |  |  |  |
| --- | --- | --- | --- |
| c | An | an | d |
| 0.24 | 0.2 | 1.05 | 1 |
| **0.24 Average** | 0.2 | 1.05 |  |
| 0.23 | 0.1 | 0.3 | 1 |
| 0.23 | 0.3 | 1.2 | 1 |
| 0.23 | 0.4 | 1.8 | 1 |
| **0.23 Average** | 0.266667 | 1.1 |  |
| 0.22 | 0.2 | 0.85 | 1 |
| 0.22 | 0.6 | 3.1 | 1 |
| **0.22 Average** | 0.4 | 1.975 |  |
| 0.21 | 0.2 | 1 | 1 |
| 0.21 | 0.4 | 1.55 | 1 |
| 0.21 | 0.5 | 2.2 | 1 |
| **0.21 Average** | 0.366667 | 1.583333 |  |
| 0.2 | 0.2 | 0.8 | 1 |
| 0.2 | 0.4 | 1.45 | 1 |
| **0.2 Average** | 0.3 | 1.125 |  |
| 0.19 | 0.2 | 0.75 | 1 |
| 0.19 | 0.3 | 1.25 | 1 |
| 0.19 | 0.4 | 1.6 | 1 |
| 0.19 | 0.6 | 1.95 | 1 |
| **0.19 Average** | 0.375 | 1.3875 |  |

Unfortunately, Gordon *et. al.* did not publish their “raw” data in this instance, rendering it difficult to determine for certain which value of *c* to use. We will, therefore, reproduce our computation values for *c* and the corresponding and :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0.24 Average** | | | 0.2 | | | 1.05 | | |
| **0.23 Average** | | | 0.266667 | | | 1.1 | | |
| **0.22 Average** | | | 0.4 | | | 1.975 | | |
| **0.21 Average** | | | 0.366667 | | | 1.583333 | | |
| **0.2 Average** | | | 0.3 | | | 1.125 | | |
| **0.19 Average** | | | 0.375 | | | 1.3875 | | |
| **0.18 Average** | | 0.375 | | | 1.3125 | | |  | |
| **0.17 Average** | 0.425 | | | 1.52725 | | |
| **0.16 Average** | 0.65 | | | 2.083333 | | |
| **0.15 Average** | 1 | | | 2.95 | | |
| **0.14 Average** | 0.55 | | | 1.45 | | |
| **0.13 Average** | 1.08 | | | 2.84 | | |
| **0.12 Average** | 1.1 | | | 2.7 | | |
| **0.11 Average** | 0.925 | | | 2.025 | | |
| **0.1 Average** | 1.15 | | | 2.175 | | |
| **Grand Average** | 0.689091 | | | 1.894709 | | |

We will simply utilize the average value for (0.689091). Also, using Excel’s sort functionality, we can determine that the range for the “raw” values of fall between 0.1 and 2.4. However, since the second-largest value is 1.8, we will test to see if this is an outlier. Using **=QUARTILE(A2:A71, 3)** and **=QUARTILE(A2:A71, 1)** we can determine that the third quartile is 1 and the first quartile is 0.4, giving us an interquartile range of 0.6. We will then calculate that and . (This test was originally proposed by John Tukey; see, for example, the below). This means that 2.4 is, in fact, an outlier but 1.8 and 0.1 are not. Therefore, we will expect the arrival rate to fall between 0.1 and 1.8 with an average of 0.689091.

Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.

Hoaglin, D. C., Iglewicz, B., & Tukey, J. W. (1986). Performance of Some Resistant Rules for Outlier Labeling. *Journal of the American Statistical Association*, *81*(396), 991–999. https://doi.org/10.1080/01621459.1986.10478363

The values for (which is the average departure rate) unfortunately do not match the data quite as well, with a range of 0.3 to 4.7 (vs. 0.15 to 1.2 in the field). A procedure similar to the above confirmed that there are no outliers in the data. Unfortunately, this suggests that this parameter should not be used. If we modify our Matlab script to allow only , has an average of 0.679592, a range of 0.17 to 1.2, an interquartile range of 0.5, and no outliers. has a range of 0.1 to 0.24, an interquartile range of 0.07, an average of 0.1579, and no outliers.

Thus, we have our final calculations for the rates: active foragers will transition to inactive foragers at a rate of approximately 0.69 with a range of 0.1 to 1.8, and inactive foragers will transition to active foragers at an average rate of 0.679592 with a range of 0.15 to 1.3.