A1: Part 2

3. Research: Twitter graph degree distributions

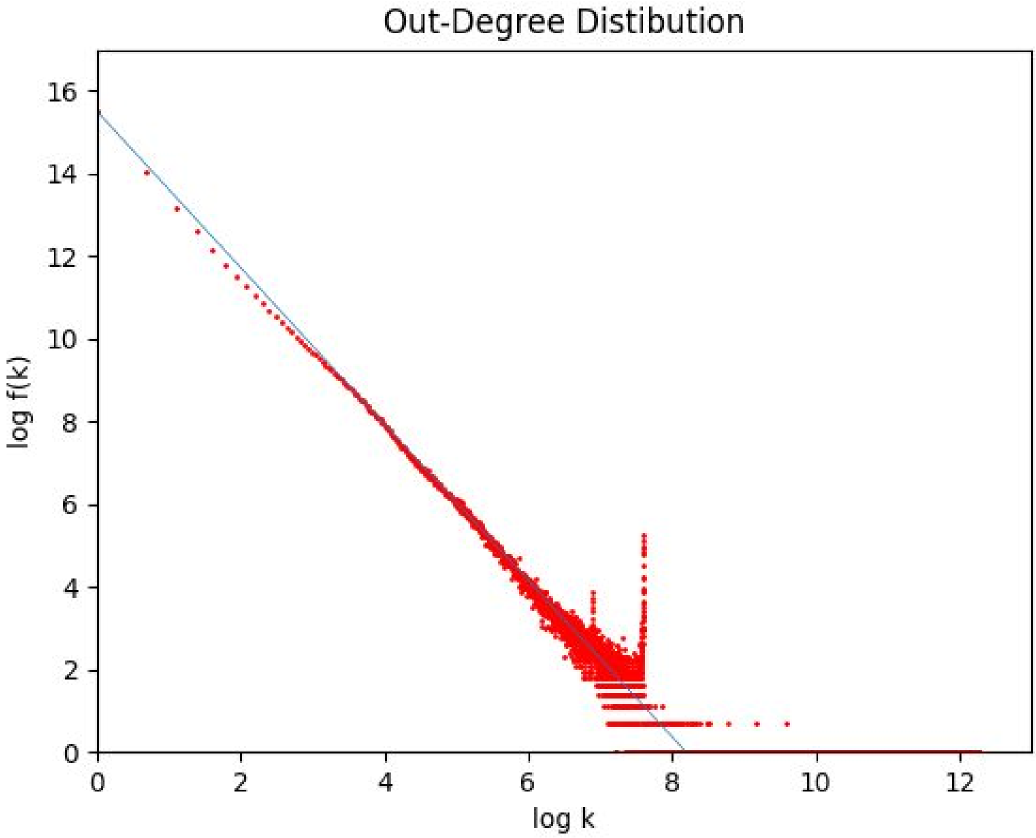
The **distribution** program computes the in and out distribution of the twitter file. Our program uses an array of size *max_degree* to store the counts of each in/out degree. So we need to pass in the *max_degree* parameter in the program, which can be computed by running the **read_blocks_seq** program from part 1 of the assignment. So the i'th element of the array will represent the count for degree i+1.

To compute the out distribution we just called the distribution program with the *records.dat* file, which was sorted by UID1. Similarly we use *merged_runs.dat* file, which is sorted by UID2, to compute the in distribution. The distribution program outputs all the (degree,count) pairs. So we just redirected the output to a text file. We used the following two commands to create two text files.

```
distribution records.dat 4096 UID1 214381 > hist_out.txt
distribution merged_runs.dat 4096 UID2 214381 > hist_in.txt
```

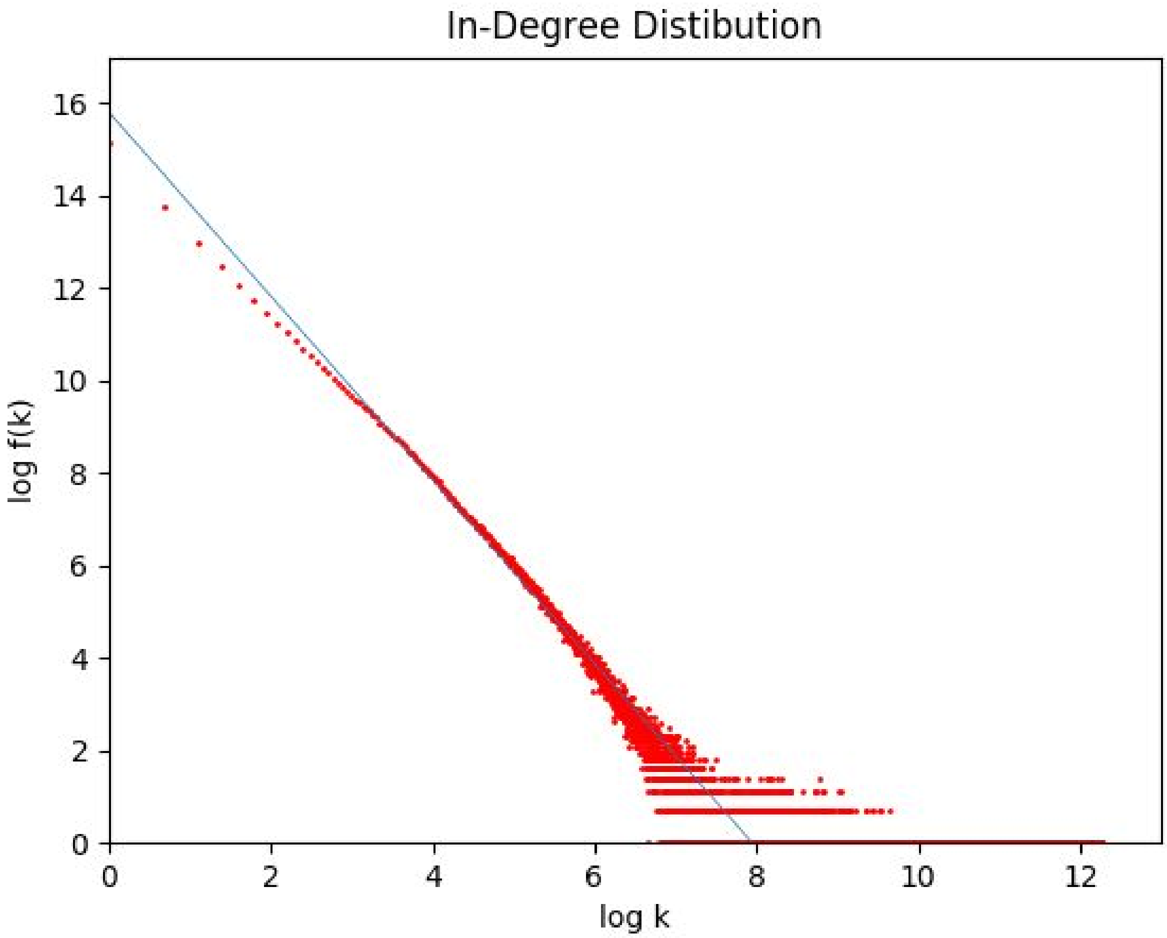
The two text files were later parsed by a python script to create one excel file called histogram.xlsx. The excel files shows both the in and out distribution computed by our distribution program.

The following log f(k) vs log k graph shows the result of out-distribution:



The red dots represent the individual (degree,count) pairs from pur experiment and the blue line is a best fit line through the points. It is clearly evident that it follows the power-law. From, the graph we can see that the y-intercept is around 15.5 and the slope is around (-15.5 / 8.2) = -1.89. Thus we can conclude that c = 1.89 and log a = 15.5

The following log f(k) vs log k graph shows the result of in-distribution:



This graph is also very similar to the out-distribution, with y-intercept at around 15.8 and the slope at around (-15.8 / 7.95) = -1.99. Thus we can conclude that c = 1.99 and log a = 15.8. So the in-distribution also follows the power-law.