Simulation of Random Graphs

Yixuan Chen

Introduction

Random graphs are models widely used to study complex systems, such as social networks, transportation networks, and biological systems. They are called random, because the vertices (or nodes) are randomly connected to each other. For example, in the Gilbert model (1959), there are N fixed points (vertices) and each edge (connection between some two vertices) is created with a fixed probability p independently of others. Gilbert observed that when p is small, the graph contains many isolated vertices, and as p becomes larger, all vertices merge into a single connected component. In my study, I aim to generate random graphs and analyze the distribution of sizes of their connected components in the range of p-values where the graph transitions from being totally disconnected to completely connected.

Method

Research Design

This research aimed to investigate the distribution of sizes of connected components of random graphs. For this study we will fix the number of vertices to be $N=100$. The probability p of joining vertices will be varied in the interval [0,1]. Every random graph is a union of several connected components. In this paper, I am interested in studying the distribution of sizes of these connected components (The size of a connected component is the number of vertices that it contains). According to American mathematician Edgar Gilbert(1959), the edges of a random graph with N vertices are paired independently with a fixed probability p, which is $G(N, p)$. As shown by Erdős, P., & Rényi (1958), if $p \gg \frac{log(N)}{N}$, the graph becomes fully connected. In this research, I focus on the values of p from 0.0 to 0.05 because I estimate the probability of the random graph becoming fully connected with size 100 will be about $\frac{log(100)}{100} \sim 0.046$.

Simulation in JavaScript

I use JavaScript as a programming language to carry out all my random graph simulations; accordingly I implemented the process of building N vertices, adding new edges, removing edges, clearing existing graphs, generating random graphs, tracing connected components, and counting the number of the component sizes for a given random graph.

Data Sampling

I first obtained data through a simulation of 1000 random graphs with N=100 vertices from p=0 to p=1 with each increment of 0.1. I found distributions are retained the same after p=0.1. Therefore, I changed my strategy to redesign my simulation process.

After I updated my strategy, I obtained all data through a simulation process of 10000 random graphs with N=100 vertices from p=0 to p=0.049. In each turn, the p increased by 0.001. The process of simulation was started from an empty histogram array. I prepared an empty 2-dimensional histogram array to store future data. The inside loop simulated 10000 random graphs with given p and cumulated the size of connected components from 0 to 100. The result above is stored as a 1-dimensional array with a given p. The outside loop simulated steps above 50 turn from p=0 to p=0.049. Those results were the 1-dimensional arrays stored in the 2-dimensional histogram array. After I had the full data of the histogram array, I downloaded it with a CSV file and imported it into R.

Statistical Analysis

I used R version 4.2.3 for all data analysis. Data import, data cleaning, data preparation, and data visualization were conducted using 'readr' , 'MASS', 'ggplot2', 'fitdistrplus', 'logspline', 'mixtools' packages.

I obtained the probability density function(PDF) for the sizes of connected components with given different p by

$$\frac{1}{nT}\sum_{x=0}^{n} xN(x)$$

N(x): the number of connected components with given size=x;

x: size;

n: total numbers of vertices.

T: the number of simulations for random graphs with n vertices.

As for a more accurate analysis of the bell shape occurring in the process, I removed zeros and outliers using the na.omit function. I picked p=0.02, 0.025, 0.03, 0.035, 0.04 to analyze the distribution of a random graph with such a condition.

Results

I estimated the distribution of different sizes of connected components from p=0.001 to p=0.050. Because there are fifty graphs and every five graphs have the same pattern of distribution, the result only shows the typical distribution with p changing.

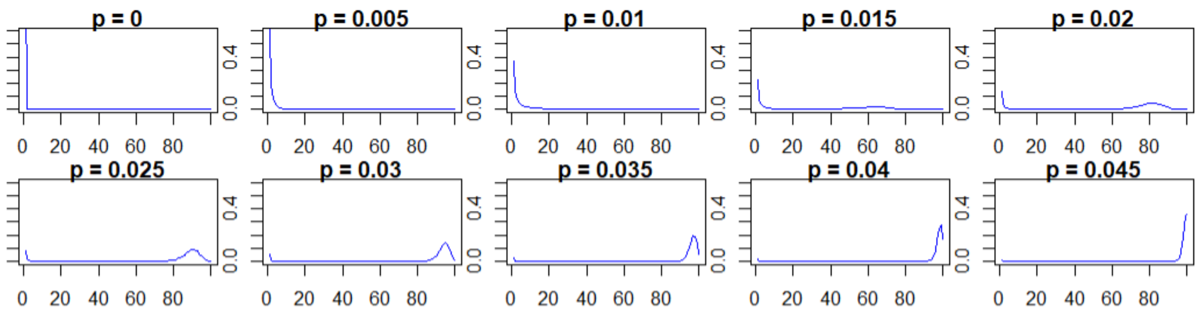


Figure 1. Distribution of sizes of connected components with different probabilities

From the graphs above, the distribution changed with p increasing. The first time a peak occurred is p=0.02. With the p increment by 0.005, the left tail was shrinking, and the peak was rising. At p=0.045, the peak disappeared and the random graph became fully connected.

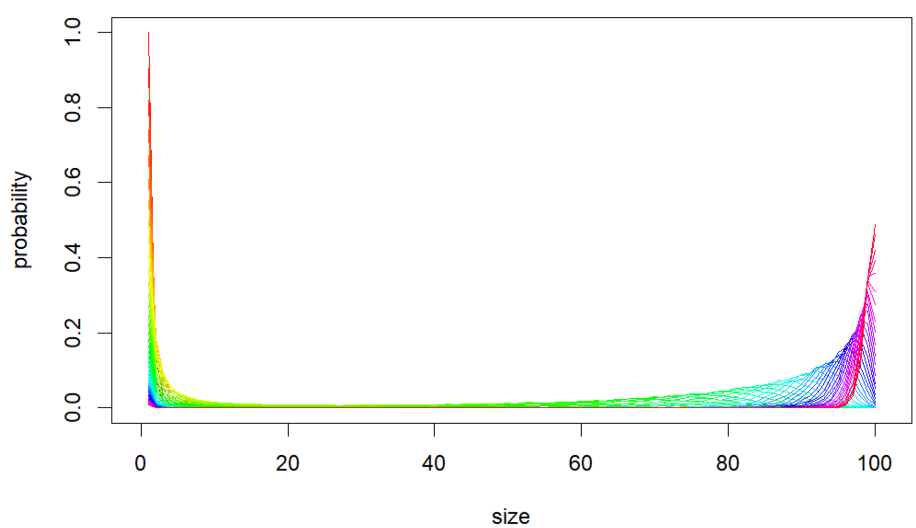


Figure 2. Distribution of connected components with probabilities from 0 to 0.05

I first fitted normal, Weibull, gamma, lognormal, and beta distributions for random graphs with p=0.02, respectively. After using fitdist function, the result showed that best performance was the beta distribution, and then the Weibull distribution was the second good fit. My result of the beta distribution is shown in Figure 3. I continued to fit those distributions with given p=0.025 by fitdist function and I found beta is still the best fit and Weibull is second best. The result shows in Figure 4.

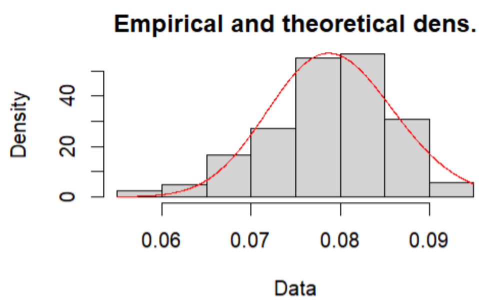


Figure 3. Beta distribution with *p*=0.02

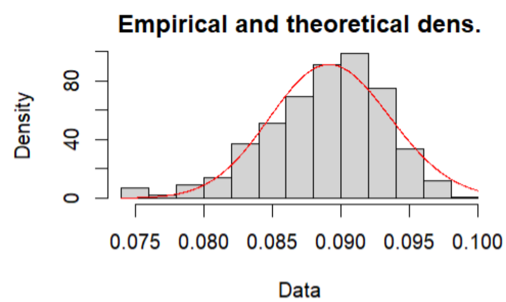


Figure 4. Beta distribution with *p*=0.025

As for probability 0.03, I found best-fit distribution is beta, and the Weibull is the second good fit. I fitted normal, Weibull, gamma, lognormal, and beta distributions for random graphs with p=0.035, respectively. As a result, the beta distribution fits the best, the second well fit is the Weibull distribution.

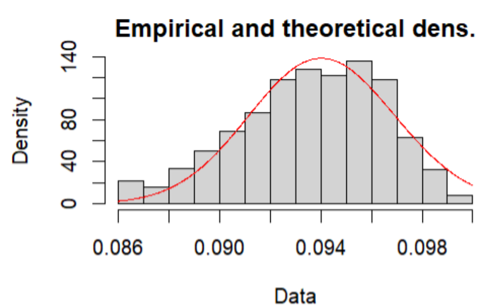


Figure 5. Beta distribution with *p*=0.03

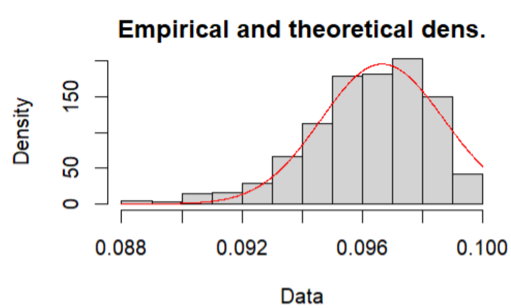


Figure 6. Beta distribution with *p*=0.035

Finally, I fit these distributions with given p=0.04, the best fit is the beta distribution, and the second best fit is still the Weibull distribution.

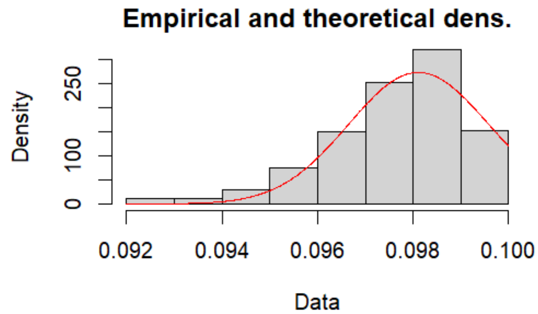


Figure 7. Beta distribution with *p*=0.04

As a result, I found no matter how the shape changed with different probabilities, the best fit is always beta distribution. The second best fit is always the Weibull distribution.

Conclusion

To sum up, the simulation of generating random graphs with *G(100, p)* confirmed the theoretical result that the random graph becomes fully connected when $p \sim \log(N)/N$. In my simulation, the p=0.045 which is about log(100)/100. At this time, random graphs became fully connected and most graphs are complete with a single connected component of size 100.

In addition, the results of probability distribution of component sizes show that beta distribution fits better than normal, gamma, lognormal, and Weibull. The Weibull fits better than normal, gamma, and lognormal. Note that beta distribution is designed for x in the interval from 0 to 1 and thus is likely to be more suitable when the component sizes are normalized by the total number of vertices, N. Other distributions may be more fitting if different scalings are used as N grows.

Limitations

This research only focuses on the number of vertices N=100. In fact, the shape of connected components of a random graph depends on both N and p. It is not entirely clear that as N becomes even larger, the component size distributions will behave similarly to what has been observed here. Therefore, I would like to explore how these distributions behave for larger values of N, e.g., N=200, N=500, and N=1000 and the values of p scaled appropriately to maintain the critical (i.e., neither totally disconnected nor completely connected) regime for the random graph.

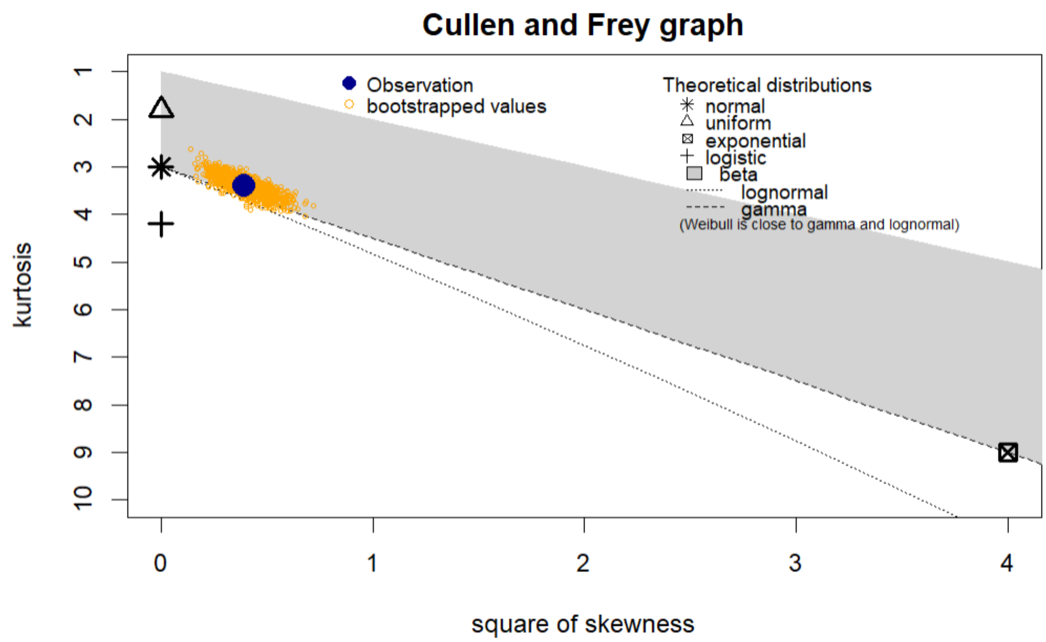
Reference

Erdős, P., & Rényi, A. (1958). On random graphs I. Dedicated to 0, Vargo, at the occasion of his 50th birthday. https://snap.stanford.edu/class/cs224w-readings/erdos59random.pdf

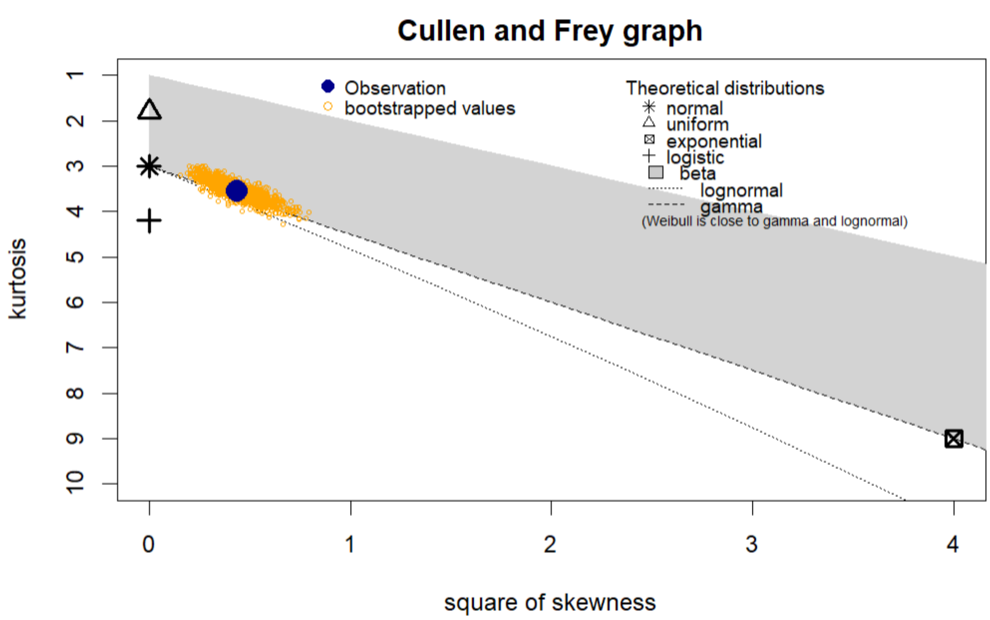
Gilbert, E. N. (1959). Random graphs. The Annals of Mathematical Statistics, 30(4), 1141-1144.

Appendix

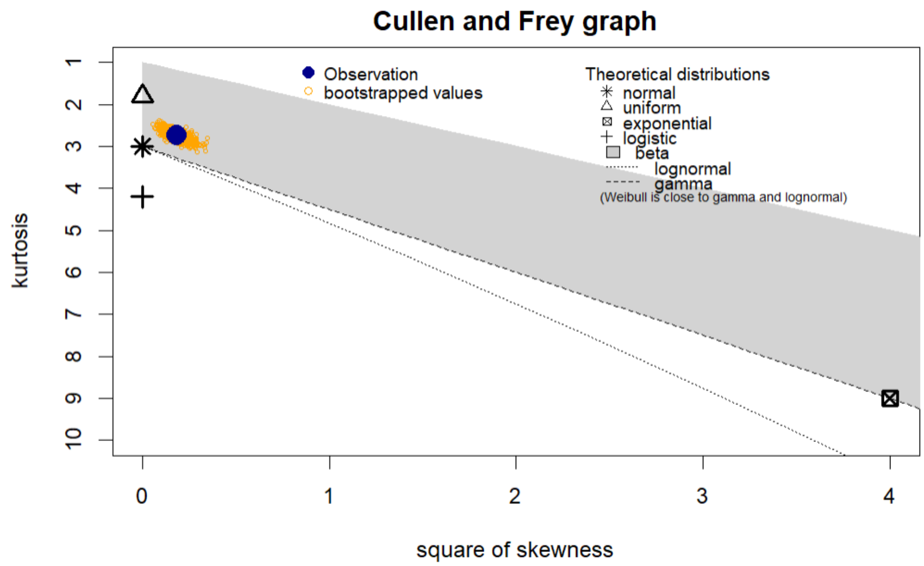
Appendix A. Result of *p*=0.02 Distribution Fit



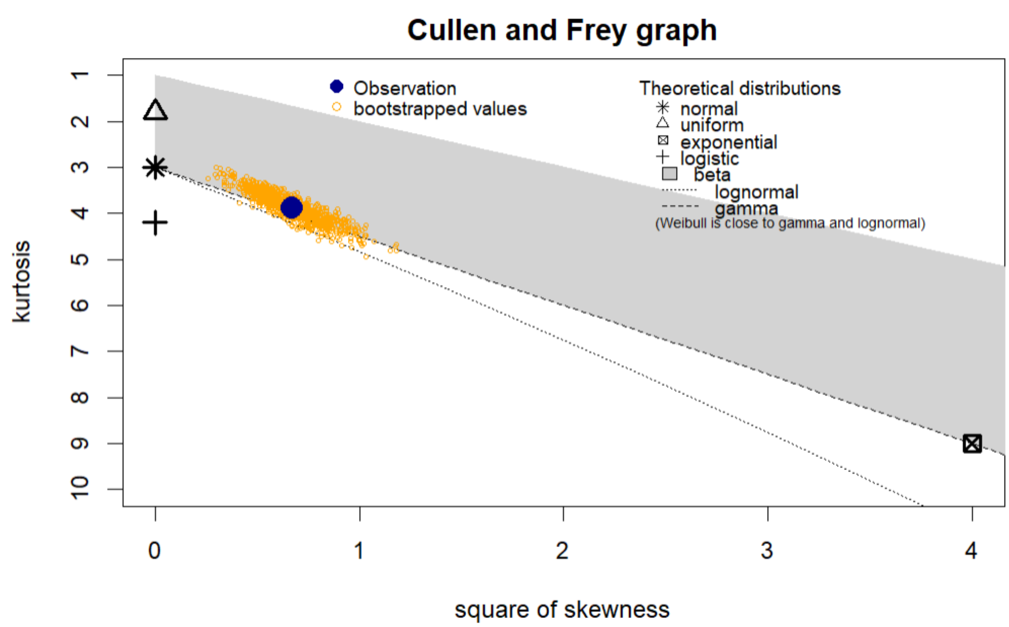
Appendix B. Result of *p*=0.025 Distribution Fit



Appendix C. Result of *p*=0.03 Distribution Fit



Appendix D. Result of *p*=0.035 Distribution Fit.



Appendix E. Result of p=0.04 Distribution Fit.