2019

Simulating and Modelling Opinion Dynamics

Jennifer Heermance
jennifer.heermance@gmail.com

ABSTRACT

SIMULATING AND MODELLING OPINION DYNAMICS

Jennifer Heermance, M.S.
Department of Statistics and Actuarial Science
Northern Illinois University, 2019
Alan Polansky, Director

The foundation of social media is conversation. Social media allows people to share ideas and opinions, as well as discuss those opinions. A point of intrigue for many social scientists is how those opinions change through interaction with others. What influences someone's opinion? When is a person willing to adapt their opinion, and when does it remain the same? Is it possible to measure these opinion dynamics? Our overall goal is to develop a more comprehensive model for opinion dynamics. The first step of this process is to simulate data that can then be analyzed and used to develop a model. When attempting to build a model for opinion dynamics in social media interactions, it is important to start small, and make sure the core of the function executes properly. Thus, we started with a small, manageable model with no error terms and only two actors. After testing this, we generalized the model and added our error terms, testing this within three different potential Ω environments. Several simulations later, we used our final model function to simulate 1000 iterations of each factor combination and return the Median Absolute deviation, proceeding to perform analysis on this data. There were no consistent factors that gave the maximum or minimum mean or standard deviation of the data. However, this does not mean there is no value in looking at these summaries. With some tweaking to the function, it would

be very possible to perform regression analysis or time series analysis to indicate not only which factors are most important, but to be able to effectively estimate the values of these factors with real world data. Additionally, there is value in experimenting with dynamic Ω matrices. In real life, various comments will affect the respect one user feels for another and will therefore change the level of influence on their opinions.

NORTHERN ILLINOIS UNIVERSITY
DE KALB, ILLINOIS

DECEMBER 2019

SIMULATING AND MODELLING OPINION DYNAMICS

BY

JENNIFER HEERMANCE

A THESIS SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE

MASTER OF SCIENCE

DEPARTMENT OF STATISTICS AND ACTUARIAL SCIENCE

Thesis Director:
Alan Polansky

ACKNOWLEDGEMENTS

Thank you to all of my friends and family who have supported me through my educational journey and beyond. I would also like to thank my professors at NIU who supported and cared for me through it all, especially Alan Polansky and Carrie Helmig.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 What is Opinion Dynamics?

The foundation of social media is conversation. Social media allows people to connect, converse, and grow through the internet. It allows people to share ideas and opinions, as well as discuss those opinions. A point of intrigue for many social scientists is how those opinions change through interaction with others. What influences someone's opinion? When is a person willing to adapt their opinion, and when does it remain the same? Is it possible to measure these changes?

1.2 Understanding Network Science

Understanding and applying opinion dynamics to social media requires a base understanding of graph theory. Network science has become an multidisciplinary approach to understanding various natural phenomena. Biologist, Sociologists, Computer Scientists, Government services, such as the Postal Service, and many more have adapted network science to analyze and expedite data in their fields (Kolacyzk, 2009, p. 3–9).

Consider a network graph with a set of vertices, V called *nodes*, connected by a set of edges, E. Elements of the set E take the form (u, v), where u and v are elements of V. Edges can be directed, meaning they go in only one direction. For example, if user u changes user v's opinion, then the edge (u, v) can be a directed edge. A graph made up of directed edges is a directed graph. Edges can also be weighted, if user u heavily influences the opinion of user v, then the *weight* of edge (u, v) may have a larger weight than other edges. Graphs can be denoted as $G = (V, E)$, where the *order* of G is the number of vertices, $N_v = |V|$, and the *size* of G is $N_e = |E|$. A conversation on social media may be a directed graph with the order equal to the number of people in the discussion, and the size dependent on the number of comments made, with varying weights. For the sake of a graphical representation, multiple edges in the same direction will not be counted.

1.3 Beginning Models and Related Work

De, Valera, Ganguly, Bhattacharya, and Rodriguez (2016) worked to create a series of models that would not only measure the changes of opinions, but would also predict how opinions would continue to evolve by focusing on a linear relationship between a user's series of opinions and the influence of other people on the user. They proposed what they call "Theorem 2", that reads as follows:

Given a collection of messages H_{t_0} recorded during a time period $[0, t_0)$ and $\lambda_u^*(t) = \mu_u$ for all $u \in G$, then,

$$\mathbb{E}_{H_t/H_{t_0}}[x^*(t)|H_{t_0}] = e^{(A\Lambda_1 - \omega I)(t-t_0)}x(t_0) + \omega(A\Lambda_1 - \omega I)^{-1}(e^{(A\Lambda_1-\omega I)(t-t_0)} - I)\alpha, \tag{1.1}$$

where $\Lambda_1 = diag[\mu]$ and $(x(t_0))_{u\in V} = \alpha_u + \Sigma_{v\in N(u)}\alpha_{uv}\Sigma_{t_i\in H_v(t_0)}d^{-\omega(t_0-t_i)}m_v(t_i)$.

This model, while using slightly different notation than we intend to use and being a bit complex, lays the groundwork for future work to be done.

Bishop (2019), Kapoor, Vergari, Rodriguez, and Valera, (2018), Amelkin, Bogdanov, and Singh (2019), and Chen, Tsaparas, Lijffijt, and De Bie, (2019), and Taibian, Gomez, De, Schölkopf, and Rodriguez (2019) also referenced De et al.'s (2016) work to develop models from different perspectives. Bishop (2019) explored these models from a Bayesian perspective. However, it was found to be very complex and difficult to continue for more than two users. Kapoor et al. (2018) worked to fix underlying problems with Bayesian nonparametric models. Amelkin et al. (2019) introduced a model to compute "Social Network Distance" between two opposing opinions. Chen et al. (2019) built a model introducing the social phenomena backfire effect and biased assimilation, and defines these phenomena as "The fact that an opposite opinion may further entrench someone in their stance, making their opinion more extreme instead of moderating it," and "The tendency of individuals to adopt other opinions if they are similar to their own", respectively.

Taibian et al. (2019) began development of consequential ranking algorithms that will fix the long term problems of current ranking algorithms and hopefully lessen the spread of misinformation and incivility. Similarly, Kim, Tabibian, Oh, Schölkop, and Gomez-Rodriguez (2018) proposed the CURB algorithm that sends articles and posts with enough "flags" to be fact-checked by a trusted third party. They found that, under certain initial conditions, the CURB algorithm can effectively reduce the spread of misinformation significantly. Another algorithm that can be used to affect the spread of misinformation, as well as predict future opinions, is the CHESHIRE algorithm, which is designed to maximize online activity by sampling the optimal times for users' incentivized actions. That is, it indicates the most influential posters that should post at peak times in order to encourage their followers to respond (Zarezade, De, Rabiee, & Rodriguez, 2017).

Minimizing the spread of misinformation is important not just in terms of who is spreading the information, but also what is being spread. Thus, being able to predict what will go viral and using that to shape the direction of opinion dynamics are crucial. Rizoiu et al. (2017) developed a Hawkes' Intensity Process that can identify not only what videos have a high potential to go viral, but also what external promotions are most effective at getting them there. Similarly, De, Bhattacharya, and Ganguly developed SMARTSHAPE, which is described as "an opinion control package that jointly selects the control users,as well as computes the optimum rate of control messages, thereby driving the networked opinion dynamics to the desired direction," (2018).

There are large scale applications of opinion dynamics and social network analysis, as well. De goes on to delve into how complex opinions are being simplified into arguments based solely on emotion are gaining in popularity, and whether or not online discussions are significantly falling into demagoguery by analyzing "voting" of opinions on various social media sites (Upadhyay, De, Pappu, & Gomez-Rodriguez, 2019). Additionally, Rashkin, Bell, Choi, and Volkova (2017) conducted a large scale study on how people all around the world respond to global events, analyzing how author word choice influences reader sentiment. It is clear that social media has a huge impact on not just individual opinions, but the framework of global societies.

1.4 Our Goals

Our overall goal is to develop a more comprehensive model for opinion dynamics. The first step of this process is to simulate data that can then be analyzed and used to develop a model. The following chapters will discuss how the simulation functions were developed, the results of various simulation studies, and the analysis of the data from the final simulations.

CHAPTER 2

SIMULATION FUNCTIONS

2.1 First Function

When attempting to build a model for opinion dynamics in social media interactions, it is important to start small, and make sure the core of the function executes properly. Thus, we considered a weighted network graph $G = (V, E)$ where $V = \mathbb{N}(2)$ and $E = \mathbb{E}(2)$. Two people are having a conversation and influencing each other's opinions by some weight. In order to simulate this, we developed a model with the following definitions:

For each vertex $v \in V$, there is a continuous stochastic process $\zeta(v, t)$, where t is time. $\Psi(t)$ is a Poisson process with rate λ. Let $\{W_k\}_{k=1}^{\infty}$ be a sequence of random variables that independently and identically are distributed following some distribution $m \in \mathbb{R}$ with support V. It is assumed that $\Psi(t)$ and $\{W_k\}_{k=1}^{\infty}$ are independent. The weights described above are given in a 2×2 matrix $\Omega_{kl} \in [0, 1]$ where $k, l \in \mathbb{N}(2)$ and $(k, l) \in \mathbb{E}(2)$.

In regards to our social media network, $\zeta(v, t)$ refers to the opinion that user v has at time t, $\Psi(t)$ regulates the times that one of the users expresses an opinion to the community V, and $\{W_k\}_{k=1}^{\infty}$ indicates which actor expressed the opinion. The matrix Ω_{kl} contains the directional influence of each user's opinion on other users' opinions. Thus, $\Omega_2 1$ is the influence of user v_2's opinion on user v_1's opinion. Additionally, it is often assumed that $\Omega_{uv} \neq \Omega_{vu}$, and $\Omega_{uu} = 0$.

For the initial model, the opinion of user u, that is, $\zeta(u,t)$ is assumed to only change under the influence of an opinion from someone else in the community to

$$\zeta(u,t) \leftarrow \Omega_{vu}\zeta(v,t) + (1-\Omega_{vu})\zeta(u,t) \tag{2.1}$$

Using this information, we used the following R-code as our first simulation:

```
sim1 <- function(lambda,Omega,m1,T,Zeta){
  tau <- rpois(1, lambda*T)
  t <- sort(runif(tau,0,T))
  w <- rbinom(tau,1,m1)+1
  z <- integer(length(w))
  for(i in 1:length(w))
    if(w[i]==1)
      Zeta[2] <- Omega[1,2]*Zeta[1]+(1-Omega[1,2])*Zeta[2]
    if(w[i]==2)
      Zeta[1] <- Omega[2,1]*Zeta[2]+(1-Omega[2,1])*Zeta[1]
    z[i] <- Zeta[w[i]]

  return(list(t=t, w=w, z=z))
}
```

where ζ is the inital values of the two user's opinions. The "sim1" function gives us a list with t, indicating the times that someone expresses an opinion, w indicating who spoke at the respective times, and $z = \zeta(w,t)$ as the final opinions of the two actors. The value $\tau \sim Poisson(1, \lambda T)$ is the number of messages that are expressed in the time interval $(0, T]$.

2.1.1 Testing Model 1

Now that we have our function, we need to make sure it is working properly. After using `set.seed(157892)`, we set the parameters $\lambda = 2$, $m_1 = .5$, $T = 1$, $\zeta = (0, 1)$, and

$$\Omega = \begin{bmatrix} 0 & .75 \\ .50 & 0 \end{bmatrix}.$$

We then obtained the following output:

```
$t
[1] 0.02170549 0.30825124 0.63838089 0.95156962
$w
[1] 2 1 2 1
$z
[1] 1.0000 0.5000 0.6250 0.5625
```

Going through the function manually allows us to make sure it is working as intended. For $i = 1, w_i = 2$, so we plug in values from the function

$$\zeta_1 \leftarrow \Omega_{21}\zeta_2 + (1 - \Omega_{21}\zeta_1$$

$$\zeta_1 \leftarrow (.5)(1) + (.5)(0)$$

$$\zeta_1 \leftarrow 0.5$$

Thus, ζ_1 has been updated from 0 to 0.5.

For $i = 2, w_i = 1$,

$$\zeta_2 \leftarrow \Omega_{12}\zeta_1 + (1 - \Omega_{12})\zeta_2$$

$$\zeta_2 \leftarrow (.75)(.5) + (.25)(1)$$

$$\zeta_2 \leftarrow 0.625$$

Thus, using the updated ζ_1, ζ_2 is now updated to 0.625.

For $i = 3, w_i = 2$,

$$\zeta_1 \leftarrow \Omega_{21}\zeta_2 + (1 - \Omega_{21}\zeta_1$$

$$\zeta_1 \leftarrow (.5)(.625) + (.5)(.5)$$

$$\zeta_1 \leftarrow 0.5625$$

Thus, ζ_1 has been re-updated to 0.5625.

Finally, for $i = 4, w_i = 1$,

$$\zeta_2 \leftarrow \Omega_{12}\zeta_1 + (1 - \Omega_{12})\zeta_2$$

$$\zeta_2 \leftarrow (.75)(.5625) + (.25)(.625)$$

$$\zeta_2 \leftarrow 0.5782$$

So the final value of ζ_2 is 0.5782.

The element z in the output shows the measure of the opinion of the person who spoke at time t. So at $t = 1$, Person 2 expressed their opinion of 1.0000. At $t = 2$, Person 1 expressed their newly adapted opinion of 0.5000, and so on. The numbers in the output match the numbers just calculated, so it is safe to say that the function, so far, is working as intended.

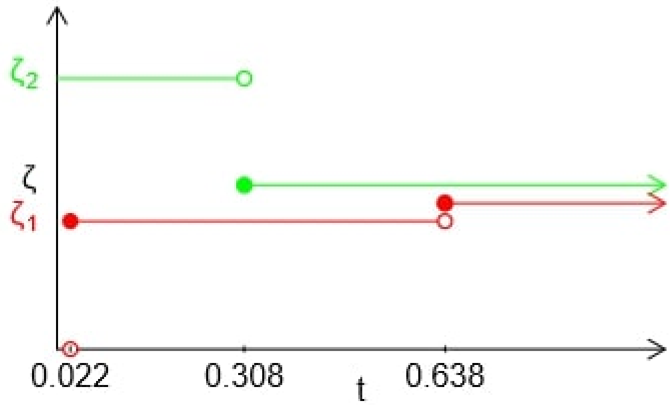

Figure 2.1: Graphical Representation of the Model I Test

2.2 Second Function

Now that we have confirmed that the basic function works, we must generalize and fully automate the process. In addition, since we are attempting to simulate real life data, we must add error terms. We built the new model from Model I with some important changes. Instead of our social network having an order of 2, it has an order of d. Thus, d people are now in the community V, and Ω is now $d \times d$. Our function will evolve to

$$\zeta(u,t) \leftarrow \Omega_{vu}\zeta(v,t) + (1-\Delta_v)(1-\Omega_{vu})\zeta(u,t) + \Delta_v(1-\Omega_{vu})U, \tag{2.2}$$

where $U \sim Beta(\gamma, \eta)$ is the weighted linear combination of the current opinion of u, the expressed opinion of v at time t, and a weighted error. $\Delta \in (0,1)$ is a constant that controls the variation of the error of the opinion. This error accounts for the possibility that user v did not express their opinion exactly.

Because in Model 1, there were only two actors, the probability of a user expressing an opinion followed a Binomial distribution with parameters $n = 1, p = m_1$. However, now that we have more than two actors, the probability of a user expressing an opinion must now follow the probability distribution $m \in \mathbb{R}^d$, which is a vector of d probabilities.

When observing data, our unknown parameters will be Ω, m, Δ, the inital parameters: ρ and ψ, the error parameters: γ and η, and λ. However, since we are simulating data, we will set each of these parameters.

2.2.1 Automation and Generalization

When generalizing Model 1, we needed to add a new argument, d, to the function, adapt Ω to be $d \times d$, and revise how w is simulated, as it can no longer be a binomial function. Our R-code for this step of building the function is as follows:

```
sim2 <- function(lambda, Omega, m, T, d, rho, psi){
  Zeta <- rbeta(d,rho,psi)
  tau <- rpois(1, lambda*T)
  t <- sort(runif(tau,0,T))
  w <- sample(1:d,tau,prob=m, replace=TRUE)
  z <- double(length(w))
  for (i in 1:tau){
    for (j in 1:d){
      if (j!=w[i]){
        Zeta[j] <- Omega[w[i],j]*Zeta[w[i]]+(1-Omega[w[i],j])*Zeta[j]
  }}
  z[i] <- Zeta[w[i]]
  }
```

```
  return(list(t=t, w=w, z=z, Zeta=Zeta))
}
```

Note that we also returned the ζ vector to ensure it is working properly.

Before we advance any further, we need to make sure the generalization and automation of this function is working as expected. We set our parameters at $\lambda = 2$, $T = 1$, $d = 6$, $\rho = 1$, $\psi = 2$, $m = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$, and

$$\Omega = \begin{bmatrix} 0 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0 \end{bmatrix}.$$

We then ran the simulation and obtained the following output:

```
sim2(lambda, Omega, m, T, d, rho, psi)
$t
[1] 0.2109341 0.5036478 0.8588514
$w
[1] 2 5 4
$z
[1] 0.03570841 0.21695110 0.20812536
$Zeta
[1] 0.2359339 0.1672276 0.2029616 0.2081254 0.2125382 0.1733964
```

Using the same process as in Section 2.1.1, we can make sure the simulation ran the functions correctly, and concur that while ζ continued to update with each message, the final vector z was constructed correctly.

2.2.2 Adding Error

Since the automation and generalization is complete, the last step is simply to add and test the error terms to our function. Adding in the rest of the parameters from (2.2), including the error parameters γ and η, our R-code for this function is as follows:

```
sim2 <- function(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta,
verbose=F) {
  zeta <- rbeta(d,rho,psi)
  tau <- max(rpois(1,lambda*T),1)
  t <- sort(runif(tau,0,T))
  w <- sample(1:d,tau,prob=m,replace=T)
  z <- matrix(0,tau,1)
  u <- rbeta(tau,gamma,eta)
  for(i in 1:tau) {
    for(j in 1:d) {

      if(verbose) {
        print(i)
        print(j)
        print(w)
        print(z)
```

```
        print(Omega)
        print(zeta)
        print(u)
        print(Delta)
      }

      if(j!=w[i]) zeta[j] <- Omega[w[i],j]*zeta[w[i]] +
          (1-Delta[w[i]])*(1-Omega[w[i],j])*zeta[j]+
          Delta[w[i]]*(1-Omega[w[i],j])*u[i]
    }
    z[i] <- zeta[w[i]]
  }
  return(list(t=t,w=w,z=z,d=d,T=T))
}
```

Note that we added the “verbose” function, so that if something did not work the way we expected, we would be able to see all of the information of the function and diagnose the problem. Additionally, the τ function is now a maximum of our Poisson distribution and 1, so that we do not get simulations with zero messages. Note, also, that we are also currently returning d and T, as this is helpful to help reference between different simulations in Chapter 3, where this model is thoroughly tested, and data from the final set of simulations will be analyzed in Chapter 4.

CHAPTER 3

SIMULATIONS

For all simulations, we will use $d = 10$ for simplicity. Additionally, three different Ω matrices will be used in these simulations, defined here:

- Free-For-All Model, where everyone affects each other equally, has weight matrix Ω_{FFA} (See Figure 3.1).

$$\Omega_{FFA} = \begin{bmatrix} 0 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0 \end{bmatrix}$$

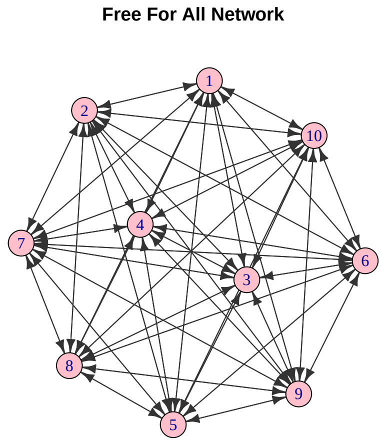

Figure 3.1: Graphical Representation of the Free For All Network

- One-Leader Model, affectionately called the Cult-Leader Model, where one person is the only influence on others, has weight matrix Ω_{CL} (See Figure 3.2).

$$\Omega_{CL} = \begin{bmatrix} 0 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

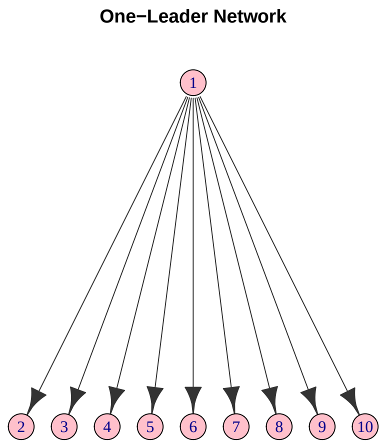

Figure 3.2: Graphical Representation of the One-Leader Network

- Two-Leader Model, affectionately called the Boy Scouts Model, where each underling eventually chooses one leader over the other, has weight matrix Ω_{BS} (See Figure 3.3).

$$\Omega_{BS} = \begin{bmatrix} 0 & 0.75 & 0.75 & 0.75 & 0.75 & 0 & 0.25 & 0.25 & 0.25 & 0.25 \\ 0 & 0 & 0.25 & 0.25 & 0.25 & 0 & 0.10 & 0.10 & 0.10 & 0.10 \\ 0 & 0.25 & 0 & 0.25 & 0.25 & 0 & 0.10 & 0.10 & 0.10 & 0.10 \\ 0 & 0.25 & 0.25 & 0 & 0.25 & 0 & 0.10 & 0.10 & 0.10 & 0.10 \\ 0 & 0.25 & 0.25 & 0.25 & 0 & 0 & 0.10 & 0.10 & 0.10 & 0.10 \\ 0 & 0.10 & 0.10 & 0.10 & 0.10 & 0 & 0.75 & 0.75 & 0.75 & 0.75 \\ 0 & 0.10 & 0.10 & 0.10 & 0.10 & 0 & 0 & 0.25 & 0.25 & 0.25 \\ 0 & 0.10 & 0.10 & 0.10 & 0.10 & 0 & 0.25 & 0 & 0.25 & 0.25 \\ 0 & 0.10 & 0.10 & 0.10 & 0.10 & 0 & 0.25 & 0.25 & 0 & 0.25 \\ 0 & 0.10 & 0.10 & 0.10 & 0.10 & 0 & 0.25 & 0.25 & 0.25 & 0 \end{bmatrix}$$

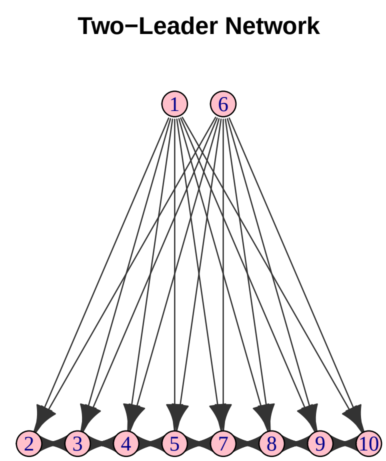

Figure 3.3: Graphical Representation of the Two-Leader Network

3.1 Plotting Function and Pilot Simulations

In order to visually see the how the simulations are working, we needed a plotting function. Thus, we added the "plotting" function, below, to run with our simulations.

```
plot.opinions <- function(x,verbose=F) {
  xl <- c(0,x$T)
  yl <- c(0,1)
  w.table <- table(factor(x$w,levels=1:d))
  plot(xl,yl,type="n",xlim=xl,ylim=yl,xlab="t",ylab="z")
  for(i in 1:x$d) {
    if(w.table[i]>0) {
      ts <- c(x$t[x$w==i])
```

```
      os <- c(x$z[x$w==i])
      if(verbose) {
        print(i)
        print(ts)
        print(os)
      }
      lines(ts,os,col=i)
    }
  }
}
```

3.1.1 Pilot Experiment Design

In order to explore what would come of our function with various parameters, we constructed an experiment with the following factors:

- $\lambda = 1$ (1 level)
- $m = (\frac{1}{10}, ..., \frac{1}{10})$ (1 level)
- $T = (10, 100)$ (2 levels)
- $d = 10$ (1 level)
- $\rho = \psi = 2$ (1 level)
- $\gamma = \eta = 2$ (1 level)
- $\Delta = (\frac{1}{4}, ..., \frac{1}{4}), (\frac{3}{4}, ..., \frac{3}{4})$ (2 levels)

- Ω = (FFA, CL, BS) (3 levels)

This gives us a total of 12 combinations, or separate simulations.

3.1.2 Results of Pilot Simulations

The specific code for each simulation can be found in Appendix C. Each simulation was run six times. Figures 3.4–3.15 show a typical result from each simulation. It was clear that when $T = 10$, we did not have enough information to see trends. The parameter, Δ, affected the plots as expected: larger Δ created larger variation in the plots. However, none of the Ω levels seemed to react too differently compared to each other, which was later discovered to be because we had programmed the transpose of the intended matrices. This was fixed for later simulations.

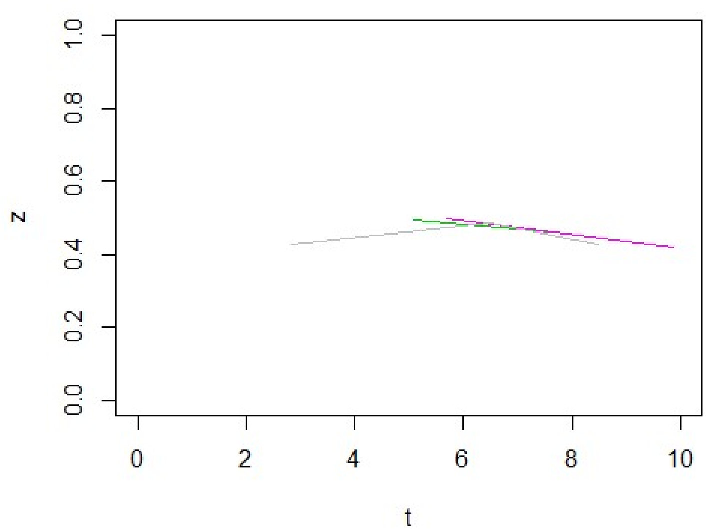

Figure 3.4: Typical Pilot Study Result for Ω_{FFA} when $T = 10$, $\Delta = (\frac{1}{4}, ..., \frac{1}{4})$

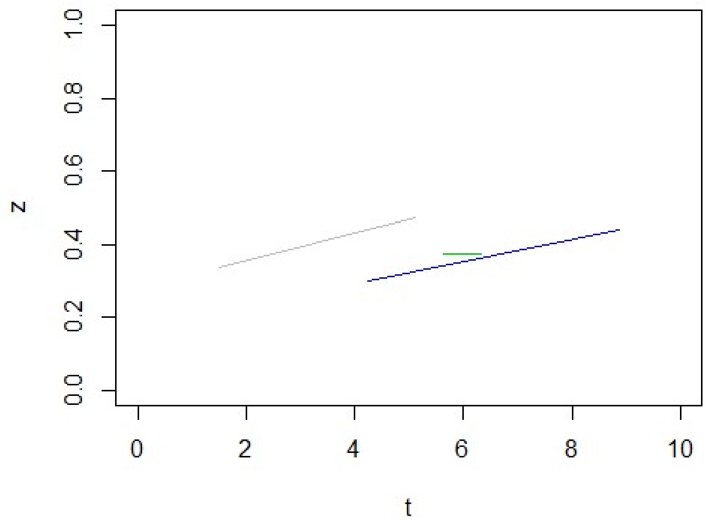

Figure 3.5: Typical Pilot Study Result for Ω_{FFA} when $T = 10$, $\Delta = (\frac{3}{4}, ..., \frac{3}{4})$

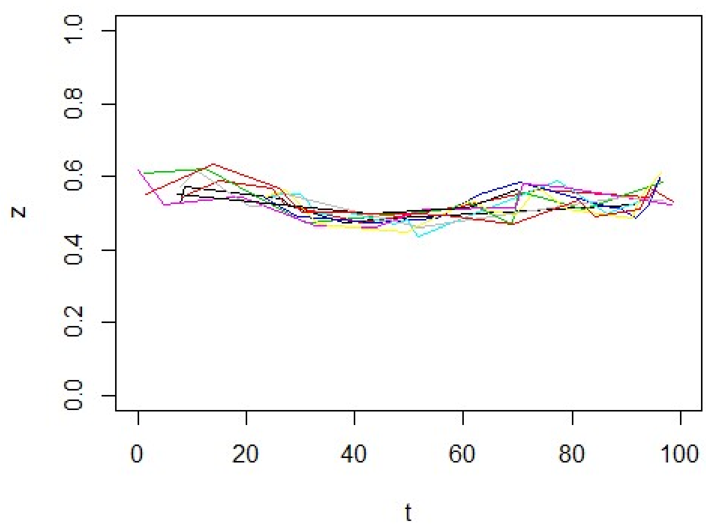

Figure 3.6: Typical Pilot Study Result for Ω_{FFA} when $T = 100$, $\Delta = (\frac{1}{4}, ..., \frac{1}{4})$

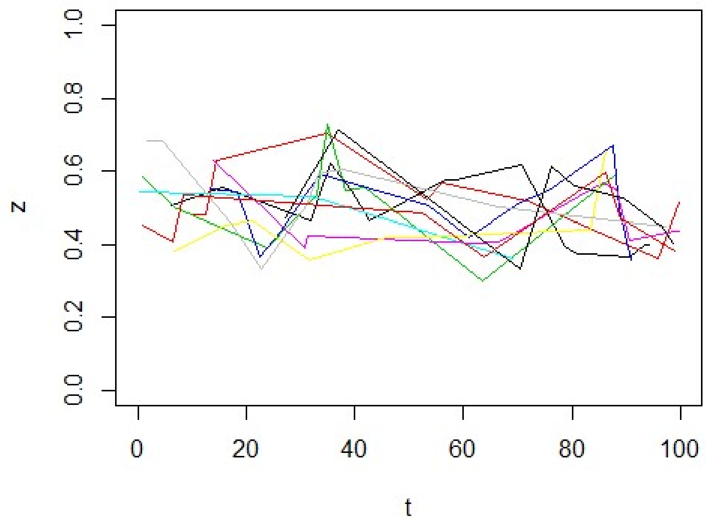

Figure 3.7: Typical Pilot Study Result for Ω_{FFA} when $T = 100$, $\Delta = (\frac{3}{4}, ..., \frac{3}{4})$

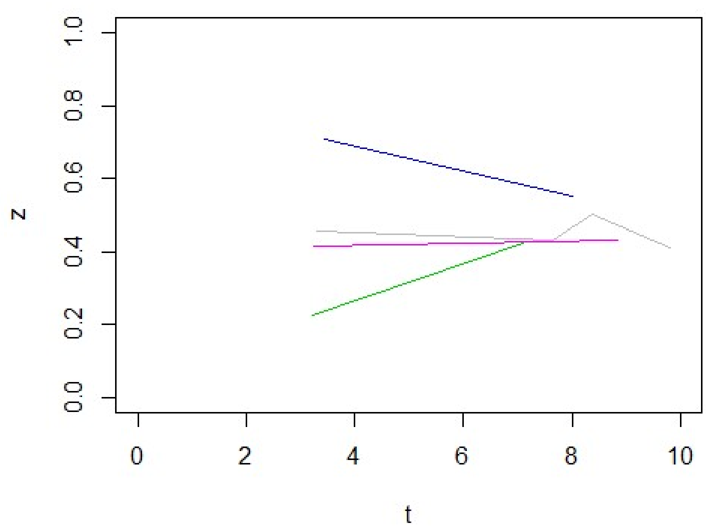

Figure 3.8: Typical Pilot Study Result for Ω_{CL} when $T = 10$, $\Delta = (\frac{1}{4}, ..., \frac{1}{4})$

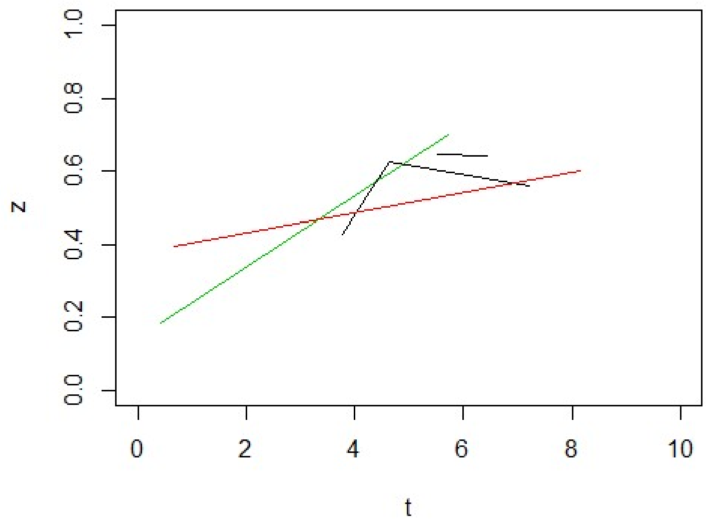

Figure 3.9: Typical Pilot Study Result for Ω_{CL} when $T = 10$, $\Delta = (\frac{3}{4}, ..., \frac{3}{4})$

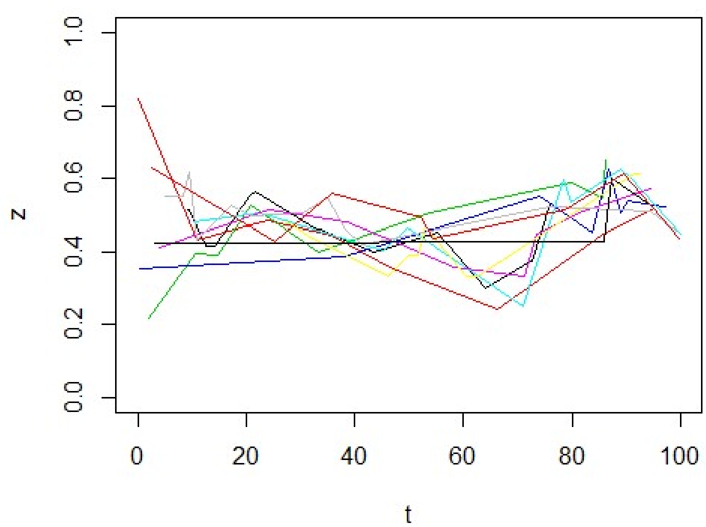

Figure 3.10: Typical Pilot Study Result for Ω_{CL} when $T = 100$, $\Delta = (\frac{1}{4}, ..., \frac{1}{4})$

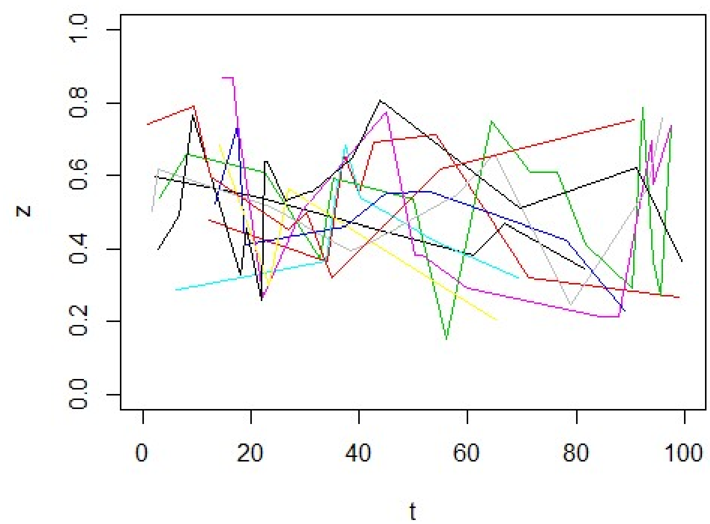

Figure 3.11: Typical Pilot Study Result for Ω_{CL} when $T = 100$, $\Delta = (\frac{3}{4}, ..., \frac{3}{4})$

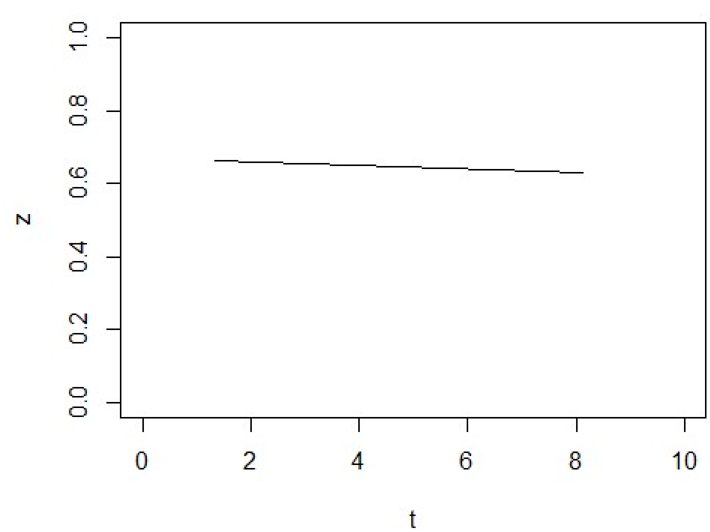

Figure 3.12: Typical Pilot Study Result for Ω_{BS} when $T = 10$, $\Delta = (\frac{1}{4}, ..., \frac{1}{4})$

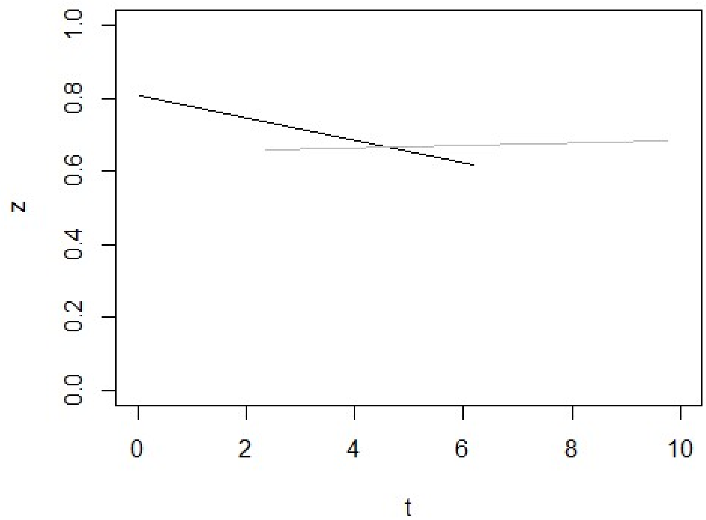

Figure 3.13: Typical Pilot Study Result for Ω_{BS} when $T = 10$, $\Delta = (\frac{3}{4}, ..., \frac{3}{4})$

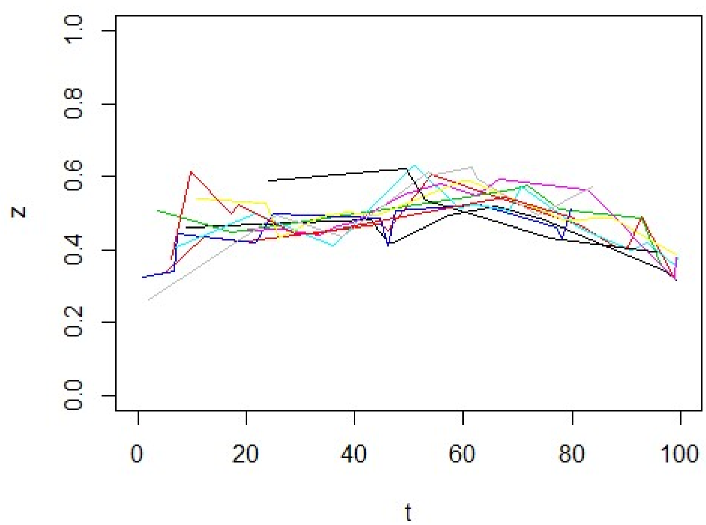

Figure 3.14: Typical Pilot Study Result for Ω_{BS} when $T = 100$, $\Delta = (\frac{1}{4}, ..., \frac{1}{4})$

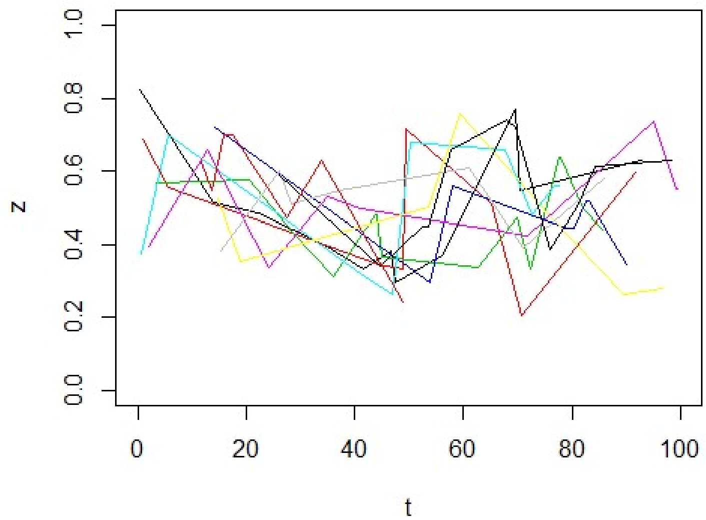

Figure 3.15: Typical Pilot Study Result for Ω_{BS} when $T = 100$, $\Delta = (\frac{3}{4}, ..., \frac{3}{4})$

3.2 Secondary Simulations

Now that it is understood that T must be large, and the Ωs were not working as expected, so we ran some simulations setting $\Delta = 0$ in order to properly see the patterns we expected from each model.

3.2.1 Secondary Experiment Design

This time, we only ran six simulations, as we wanted to continue to test values of T and observe the behaviors of each Ω model without any error terms. The design is as follows:

- $\lambda = 1$ (1 level)

- $m = (\frac{1}{10}, ..., \frac{1}{10})$ (1 level)
- $T = (100, 1000)$ (2 levels)
- $d = 10$ (1 level)
- $\rho = \psi = 2$ (1 level)
- $\gamma = \eta = 2$ (1 level)
- $\Delta = 0$ (1 level)
- $\Omega =$ (FFA, CL, BS) (3 levels)

3.2.2 Results of Secondary Simulations

Figures 3.16–3.21 show a typical result from each simulation. Thankfully, each model acted as expected without the error terms included, so the function appears to be set up correctly. Additionally, it is clear that T does not need to be very large to see the patterns emerge. The patterns emerged well as $T = 100$.

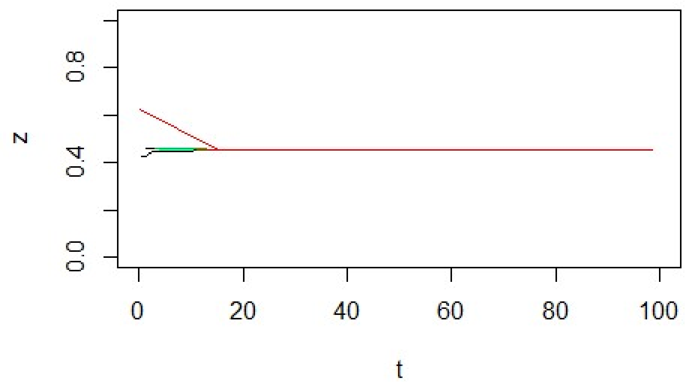

Figure 3.16: Typical Secondary Study Result for Ω_{FFA} when $T = 100$

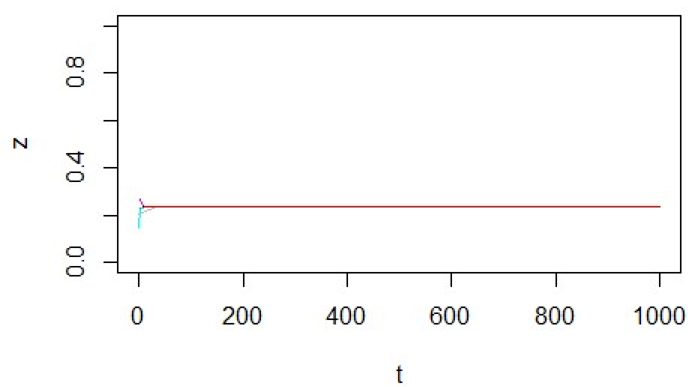

Figure 3.17: Typical Secondary Study Result for Ω_{FFA} when $T = 1000$

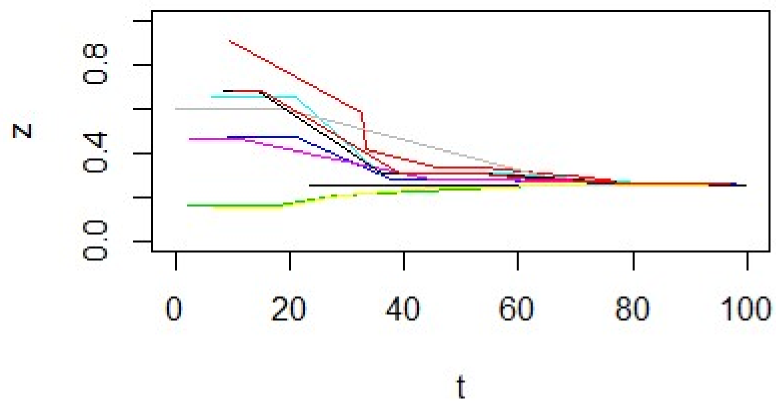

Figure 3.18: Typical Secondary Study Result for Ω_{CL} when $T = 100$

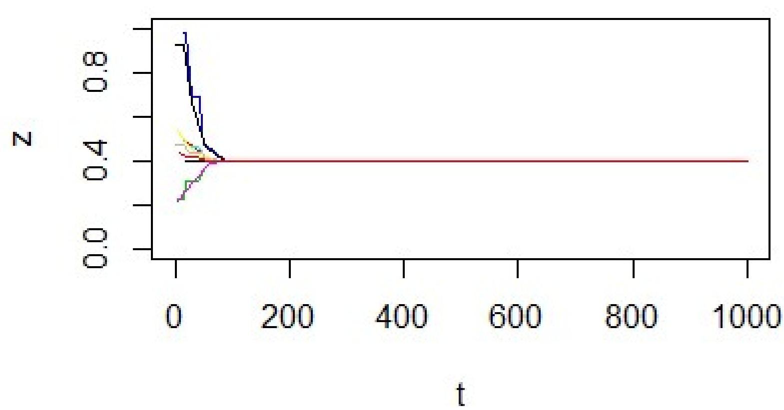

Figure 3.19: Typical Secondary Study Result for Ω_{CL} when $T = 1000$

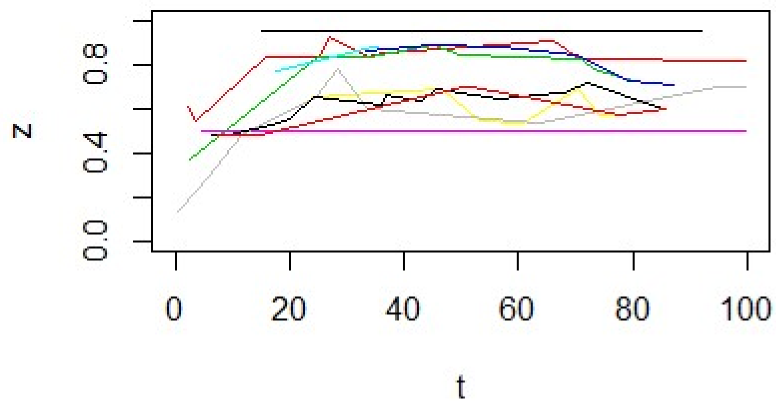

Figure 3.20: Typical Secondary Study Result for Ω_{BS} when $T = 100$

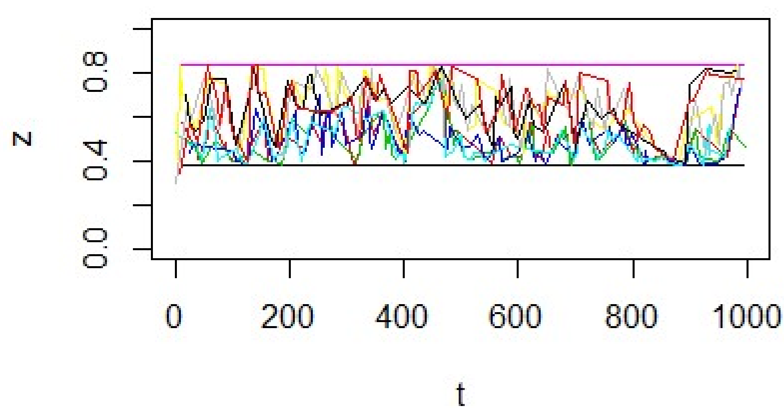

Figure 3.21: Typical Secondary Study Result for Ω_{BS} when $T = 1000$

3.3 Final Function and Simulations

Now that we are certain our function works properly, our next step is to use the function to simulate data to analyze. For this purpose, we use one last function to gather and simulate the data:

```
simulation <- function(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, b=1000) {
  a <- vector(mode="double",length=b)
  for(i in 1:b) a[i] <- mad(sim2(lambda,Omega,m,Delta,T,d,rho,psi,gamma,eta)$zeta)
  return(a)
}
```

This “simulation” function runs our “sim2” function 1000 times and returns the Median Absolute Deviation of each ”sim2” run in a vector of length 1000.

3.3.1 Final Experiment Design

For this set of simulations, we not only want to get information from the different Ω matrices, but we also want to look at when m is not always uniform. After all, some influencers are much more talkative than the influencees. We will define these new probability distributions below and denote them as “chatty”. Additionally, we want to get information from three different levels of Δ, and explore three different combinations of λ and T, as they essentially work together to do the same thing. Finally, we want to explore what happens when ρ and ψ are equally 1 and 2, and when γ and η are equally 1 and 2. These factors and their levels are below:

- $\Omega = (FFA_{unif}, CL_{unif}, CL_{chatty}, BS_{unif}, BS_{chatty})$ (5 levels)

  - unif: m is uniform
  - $CL_{chatty} : m = (\frac{1}{2}, \frac{1}{18}, ..., \frac{1}{18})$
  - $BS_{chatty} : m = (\frac{1}{4}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{4}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16})$

- $\Delta = \{(\frac{1}{4}, ..., \frac{1}{4}), (\frac{1}{2}, ..., \frac{1}{2}), (\frac{3}{4}, ..., \frac{3}{4})\}$ (3 levels)
- $\gamma = \eta = (1, 1), (2, 2)$ (2 levels)
- $\rho = \psi = (1, 1), (2, 2)$ (2 levels)
- $(\lambda, T) = \{(1, 100), (2, 100), (2, 150)\}$ (3 levels)
  - There are several different possible levels to choose
  - In the past, we worked with $\lambda = 1$ and $T = (10, 100, 1000)$
- $d = 10$ (1 level)

This design gives us a total of 180 different runs. Additionally, since we are collecting this data, each run needs to be stored in its own unique object. Thus, the name of each object will follow this format:

```
Omega.m.Delta.rhopsi.gammaeta.lamba*T
```

For example, the simulation that runs the Free for all model where m is uniform, where $\Delta = .25$, $\gamma = \eta = 1$, $\rho = \psi = 1$, and the product of λ and T is 100 (that is, $\lambda = 1, T = 100$, the name of the object will be:

```
FFA.unif.25.1.1.100
```

The code for each of these simulations can be found in Appendix C. The analysis of this data will take place in Chapter 4.

CHAPTER 4
DATA ANALYSIS

Our first step was getting our data into a data frame, where the first seven columns match the first seven columns of the table in Appendix A, the next 1000 columns are the 1000 simulations from each combination, and then the last two columns are the mean and standard deviation of each combination. We also added on a column called λT to make analysis easier.

First and foremost, we looked at and plotted the summary statistics for all of the data. Figures 4.1 and 4.2 show that the means and standard deviations already are approximately normal. However, our goals are to see if certain factors seem to influence the mean and standard deviation of the MAD more than others. With five different Ω environments, we should restrict our focus to each of those environments separately. After filtering the data for each level, that is, $\lambda \times T = 100, 200, 300$, $\rho = \psi = 1, 2$, $\gamma = \eta = 1, 2$, and $\delta = .25, .50, .75$, we looked at the means for each factor within each Ω. This is summarized in Appendix B.

Laying it out like this, however, does not seem to indicate any clear linear trends among the factors. Increasing δ does not consistently increase the mean or standard deviation of our simulations, nor are there consistent patterns among the other factors. There also were no consistent factors that gave the maximum or minimum mean or standard deviation. However, this does not mean there is no value in looking at these summaries. With some tweaking to the function, it would be very possible to perform regression analysis or time series analysis to indicate not only which factors are most important, but to be able to effectively estimate the values of these factors with real world data. Additionally, there is value in experimenting with dynamic Ω matrices. In real life, various comments will affect the respect one user feels for another and will therefore change the level of influence on their opinions.

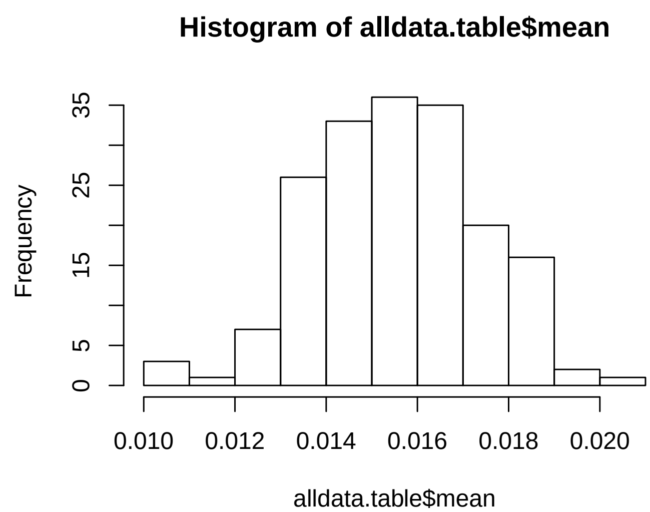

Figure 4.1: Histogram of "Mean" Column

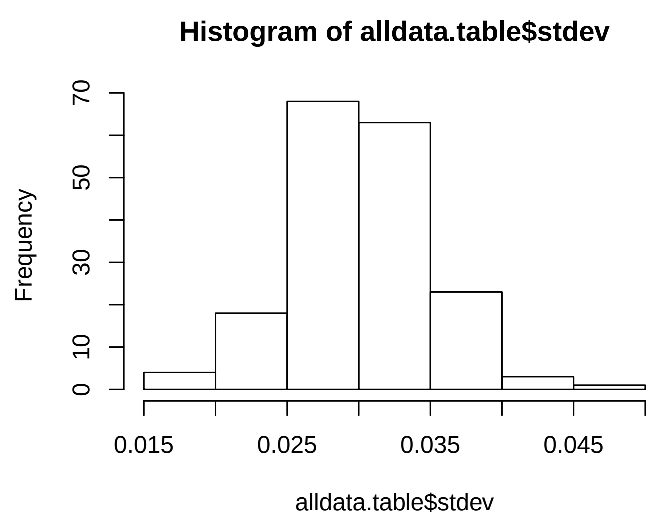

Figure 4.2: Histogram of "Stdev" Column

REFERENCES

Amelkin, V., Bogdanov, P., & Singh, A. K. (2019). A distance measure for the analysis of polar opinion dynamics in social networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *13*(4), 38.

Bishop, J. L. (2019). *Exploring a bayesian analysis of opinion dynamics using the approximate bayesian computation method* (Unpublished doctoral dissertation). Northern Illinois University.

Chen, X., Tsaparas, P., Lijffijt, J., & De Bie, T. (2019). Opinion dynamics with backfire effect and biased assimilation. *arXiv preprint arXiv:1903.11535*.

De, A., Bhattacharya, S., & Ganguly, N. (2018). Shaping opinion dynamics in social networks. In *Proceedings of the 17th international conference on autonomous agents and multiagent systems* (pp. 1336–1344).

De, A., Valera, I., Ganguly, N., Bhattacharya, S., & Rodriguez, M. G. (2016). Learning and forecasting opinion dynamics in social networks. In *Advances in neural information processing systems* (pp. 397–405).

Kapoor, J., Vergari, A., Rodriguez, M. G., & Valera, I. (2018). Bayesian nonparametric hawkes processes. In *All of bayesian nonparametrics: Especially the useful bits (bnp@ neurips 2018).*

Kim, J., Tabibian, B., Oh, A., Schölkopf, B., & Gomez-Rodriguez, M. (2018). Leveraging the crowd to detect and reduce the spread of fake news and misinformation. In *Proceedings of the eleventh acm international conference on web search and data mining* (pp. 324–332).

Kolacyzk. (2009). *Statistical analysis of network data*. Springer.

Rashkin, H., Bell, E., Choi, Y., & Volkova, S. (2017). Multilingual connotation frames: A case study on social media for targeted sentiment analysis and forecast. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 459–464).

Rizoiu, M.-A., Xie, L., Sanner, S., Cebrian, M., Yu, H., & Van Hentenryck, P. (2017). Expecting to be hip: Hawkes intensity processes for social media popularity. In *Proceedings of the 26th international conference on world wide web* (pp. 735–744).

Tabibian, B., Gómez, V., De, A., Schölkopf, B., & Rodriguez, M. G. (2019). Consequential ranking algorithms and long-term welfare. *arXiv preprint arXiv:1905.05305*.

Upadhyay, U., De, A., Pappu, A., & Gomez-Rodriguez, M. (2019). On the complexity of opinions and online discussions. In *Proceedings of the twelfth acm international conference on web search and data mining* (pp. 258–266).

Zarezade, A., De, A., Rabiee, H., & Rodriguez, M. G. (2017). Cheshire: An online algorithm for activity maximization in social networks. *arXiv preprint arXiv:1703.02059*.

APPENDIX A

FINAL SIMULATION DATA TABLE

Omega	m	δ	ρ/ψ	γ/η	λ	T	Mean	SD
FFA	unif	.25	1	1	1	100	0.01454302	0.02673883
FFA	unif	.25	1	1	2	100	0.01388799	0.02770090
FFA	unif	.25	1	1	2	150	0.01638925	0.03238470
FFA	unif	.25	1	2	1	100	0.01518766	0.02867308
FFA	unif	.25	1	2	2	100	0.01485499	0.02767786
FFA	unif	.25	1	2	2	150	0.01430214	0.02974353
FFA	unif	.25	2	1	1	100	0.01296929	0.02430849
FFA	unif	.25	2	1	2	100	0.01615451	0.02673086
FFA	unif	.25	2	1	2	150	0.01569937	0.02980309
FFA	unif	.25	2	2	1	100	0.01224023	0.02122648
FFA	unif	.25	2	2	2	100	0.01536284	0.02727210
FFA	unif	.25	2	2	2	150	0.01648091	0.03072176
FFA	unif	.50	1	1	1	100	0.01690911	0.03241352
FFA	unif	.50	1	1	2	100	0.01323209	0.02516987
FFA	unif	.50	1	1	2	150	0.01647333	0.02958221
FFA	unif	.50	1	2	1	100	0.01727051	0.03218182
FFA	unif	.50	1	2	2	100	0.01474823	0.03312846
FFA	unif	.50	1	2	2	150	0.01579271	0.03469238
FFA	unif	.50	2	1	1	100	0.01276101	0.02780042
FFA	unif	.50	2	1	2	100	0.01372849	0.02860884
FFA	unif	.50	2	1	2	150	0.01633663	0.03264018
FFA	unif	.50	2	2	1	100	0.01562371	0.03077855
FFA	unif	.50	2	2	2	100	0.01770357	0.03443705
FFA	unif	.50	2	2	2	150	0.01049299	0.01679836

Omega	m	δ	ρ/ψ	γ/η	λ	T	Mean	SD
FFA	unif	.75	1	1	1	100	0.02082839	0.04463825
FFA	unif	.75	1	1	2	100	0.01492636	0.02690018
FFA	unif	.75	1	1	2	150	0.01539921	0.02989589
FFA	unif	.75	1	2	1	100	0.01795594	0.03757156
FFA	unif	.75	1	2	2	100	0.01423999	0.02704218
FFA	unif	.75	1	2	2	150	0.01429996	0.02840973
FFA	unif	.75	2	1	1	100	0.01479011	0.02572038
FFA	unif	.75	2	1	2	100	0.01742796	0.03353921
FFA	unif	.75	2	1	2	150	0.01372433	0.02806426
FFA	unif	.75	2	2	1	100	0.01604239	0.02910034
FFA	unif	.75	2	2	2	100	0.01675429	0.03327940
FFA	unif	.75	2	2	2	150	0.01812025	0.03626262
CL	unif	.25	1	1	1	100	0.01629643	0.03181985
CL	unif	.25	1	1	2	100	0.01386177	0.02845265
CL	unif	.25	1	1	2	150	0.01681246	0.03666013
CL	unif	.25	1	2	1	100	0.01605532	0.03705429
CL	unif	.25	1	2	2	100	0.01797572	0.03434546
CL	unif	.25	1	2	2	150	0.01613155	0.03307921
CL	unif	.25	2	1	1	100	0.01470992	0.03087078
CL	unif	.25	2	1	2	100	0.01562345	0.02678667
CL	unif	.25	2	1	2	150	0.01372735	0.02700095
CL	unif	.25	2	2	1	100	0.01718722	0.03656949
CL	unif	.25	2	2	2	100	0.01801361	0.03768052
CL	unif	.25	2	2	2	150	0.01475166	0.02611991

Omega	m	δ	ρ/ψ	γ/η	λ	T	Mean	SD
CL	unif	.50	1	1	1	100	0.01419811	0.02855016
CL	unif	.50	1	1	2	100	0.01781553	0.03176944
CL	unif	.50	1	1	2	150	0.01463842	0.03191392
CL	unif	.50	1	2	1	100	0.01591431	0.03202770
CL	unif	.50	1	2	2	100	0.01624696	0.03713887
CL	unif	.50	1	2	2	150	0.01373555	0.02571016
CL	unif	.50	2	1	1	100	0.01356281	0.02214253
CL	unif	.50	2	1	2	100	0.01859127	0.03651030
CL	unif	.50	2	1	2	150	0.01768657	0.03576844
CL	unif	.50	2	2	1	100	0.01542574	0.03010946
CL	unif	.50	2	2	2	100	0.01382568	0.02222357
CL	unif	.50	2	2	2	150	0.01881939	0.03656844
CL	unif	.75	1	1	1	100	0.01919868	0.04141858
CL	unif	.75	1	1	2	100	0.01415836	0.02333868
CL	unif	.75	1	1	2	150	0.01461923	0.03072932
CL	unif	.75	1	2	1	100	0.01306488	0.02297343
CL	unif	.75	1	2	2	100	0.01801297	0.03658911
CL	unif	.75	1	2	2	150	0.01478384	0.02736467
CL	unif	.75	2	1	1	100	0.01679732	0.03238756
CL	unif	.75	2	1	2	100	0.01458912	0.02561660
CL	unif	.75	2	1	2	150	0.01391274	0.02841666
CL	unif	.75	2	2	1	100	0.01527848	0.03268936
CL	unif	.75	2	2	2	100	0.01424511	0.02451667
CL	unif	.75	2	2	2	150	0.01800600	0.03455285

Omega	m	δ	ρ/ψ	γ/η	λ	T	Mean	SD
CL	chatty	.25	1	1	1	100	0.01554775	0.03384612
CL	chatty	.25	1	1	2	100	0.01605880	0.03272508
CL	chatty	.25	1	1	2	150	0.01311797	0.02247006
CL	chatty	.25	1	2	1	100	0.01883470	0.03334317
CL	chatty	.25	1	2	2	100	0.01505577	0.02640447
CL	chatty	.25	1	2	2	150	0.01699349	0.03482672
CL	chatty	.25	2	1	1	100	0.01462606	0.02615928
CL	chatty	.25	2	1	2	100	0.01610109	0.03415547
CL	chatty	.25	2	1	2	150	0.01638574	0.02822430
CL	chatty	.25	2	2	1	100	0.01663347	0.03467032
CL	chatty	.25	2	2	2	100	0.01510184	0.03040234
CL	chatty	.25	2	2	2	150	0.01460022	0.02523054
CL	chatty	.50	1	1	1	100	0.01295817	0.01973940
CL	chatty	.50	1	1	2	100	0.01317667	0.02342864
CL	chatty	.50	1	1	2	150	0.01579691	0.02891538
CL	chatty	.50	1	2	1	100	0.01436732	0.02612510
CL	chatty	.50	1	2	2	100	0.01422019	0.02712746
CL	chatty	.50	1	2	2	150	0.01704504	0.03526218
CL	chatty	.50	2	1	1	100	0.01587279	0.03256601
CL	chatty	.50	2	1	2	100	0.01564592	0.02834725
CL	chatty	.50	2	1	2	150	0.01592519	0.03169274
CL	chatty	.50	2	2	1	100	0.01742273	0.03211754
CL	chatty	.50	2	2	2	100	0.01829984	0.03314179
CL	chatty	.50	2	2	2	150	0.01146731	0.02072743

Omega	m	δ	ρ/ψ	γ/η	λ	T	Mean	SD
CL	chatty	.75	1	1	1	100	0.01448837	0.02840036
CL	chatty	.75	1	1	2	100	0.01521556	0.02809303
CL	chatty	.75	1	1	2	150	0.01600920	0.03472977
CL	chatty	.75	1	2	1	100	0.01539344	0.03071289
CL	chatty	.75	1	2	2	100	0.01751795	0.02987664
CL	chatty	.75	1	2	2	150	0.01420497	0.02907827
CL	chatty	.75	2	1	1	100	0.01530034	0.02435903
CL	chatty	.75	2	1	2	100	0.01341488	0.02309610
CL	chatty	.75	2	1	2	150	0.01302211	0.02249854
CL	chatty	.75	2	2	1	100	0.01666925	0.03319576
CL	chatty	.75	2	2	2	100	0.01601620	0.03091135
CL	chatty	.75	2	2	2	150	0.01632381	0.03309020
BS	unif	.25	1	1	1	100	0.01737860	0.03521266
BS	unif	.25	1	1	2	100	0.01592871	0.02934203
BS	unif	.25	1	1	2	150	0.01863985	0.04569636
BS	unif	.25	1	2	1	100	0.01458565	0.02733987
BS	unif	.25	1	2	2	100	0.01832494	0.03448064
BS	unif	.25	1	2	2	150	0.01479358	0.02899447
BS	unif	.25	2	1	1	100	0.01564855	0.02537430
BS	unif	.25	2	1	2	100	0.01499163	0.02502683
BS	unif	.25	2	1	2	150	0.01472895	0.02633485
BS	unif	.25	2	2	1	100	0.01406378	0.02641973
BS	unif	.25	2	2	2	100	0.01638000	0.03558556
BS	unif	.25	2	2	2	150	0.01459578	0.02657785

Omega	m	δ	ρ/ψ	γ/η	λ	T	Mean	SD
BS	unif	.50	1	1	1	100	0.01329262	0.02615025
BS	unif	.50	1	1	2	100	0.01567759	0.03043725
BS	unif	.50	1	1	2	150	0.01606999	0.02926664
BS	unif	.50	1	2	1	100	0.01364537	0.02417214
BS	unif	.50	1	2	2	100	0.01566691	0.02827616
BS	unif	.50	1	2	2	150	0.01542061	0.03189097
BS	unif	.50	2	1	1	100	0.01081021	0.01698563
BS	unif	.50	2	1	2	100	0.01522161	0.02811493
BS	unif	.50	2	1	2	150	0.01793923	0.03459891
BS	unif	.50	2	2	1	100	0.01477002	0.02686053
BS	unif	.50	2	2	2	100	0.01608556	0.02955255
BS	unif	.50	2	2	2	150	0.01312274	0.02163897
BS	unif	.75	1	1	1	100	0.01557121	0.03057171
BS	unif	.75	1	1	2	100	0.01811338	0.03619066
BS	unif	.75	1	1	2	150	0.01555971	0.03439283
BS	unif	.75	1	2	1	100	0.01551535	0.03179610
BS	unif	.75	1	2	2	100	0.01785715	0.03676507
BS	unif	.75	1	2	2	150	0.01931871	0.03800677
BS	unif	.75	2	1	1	100	0.01648772	0.03156283
BS	unif	.75	2	1	2	100	0.01865516	0.03590721
BS	unif	.75	2	1	2	150	0.01270252	0.02218658
BS	unif	.75	2	2	1	100	0.01760297	0.03353963
BS	unif	.75	2	2	2	100	0.01715870	0.03397052
BS	unif	.75	2	2	2	150	0.01343938	0.02775247

Omega	m	δ	ρ/ψ	γ/η	λ	T	Mean	SD
BS	chatty	.25	1	1	1	100	0.01359627	0.02543073
BS	chatty	.25	1	1	2	100	0.01638461	0.03561493
BS	chatty	.25	1	1	2	150	0.01426931	0.02700739
BS	chatty	.25	1	2	1	100	0.01893120	0.04021732
BS	chatty	.25	1	2	2	100	0.01380119	0.03165268
BS	chatty	.25	1	2	2	150	0.01525835	0.02863771
BS	chatty	.25	2	1	1	100	0.01576950	0.03102530
BS	chatty	.25	2	1	2	100	0.01616844	0.03201693
BS	chatty	.25	2	1	2	150	0.01277599	0.02902737
BS	chatty	.25	2	2	1	100	0.01597729	0.02924047
BS	chatty	.25	2	2	2	100	0.01319539	0.02476358
BS	chatty	.25	2	2	2	150	0.01623045	0.03063288
BS	chatty	.50	1	1	1	100	0.01445068	0.02904297
BS	chatty	.50	1	1	2	100	0.01712765	0.03158953
BS	chatty	.50	1	1	2	150	0.01350165	0.02527270
BS	chatty	.50	1	2	1	100	0.01744075	0.03677151
BS	chatty	.50	1	2	2	100	0.01652964	0.03032328
BS	chatty	.50	1	2	2	150	0.01675164	0.03400548
BS	chatty	.50	2	1	1	100	0.01692634	0.03838347
BS	chatty	.50	2	1	2	100	0.01516608	0.02613665
BS	chatty	.50	2	1	2	150	0.01364317	0.02656073
BS	chatty	.50	2	2	1	100	0.01516805	0.02605669
BS	chatty	.50	2	2	2	100	0.01868959	0.03487781
BS	chatty	.50	2	2	2	150	0.01692923	0.03489425

Omega	m	δ	ρ/ψ	γ/η	λ	T	Mean	SD
BS	chatty	.75	1	1	1	100	0.01818425	0.03408384
BS	chatty	.75	1	1	2	100	0.01085851	0.01756177
BS	chatty	.75	1	1	2	150	0.01741027	0.03514766
BS	chatty	.75	1	2	1	100	0.01792958	0.03641161
BS	chatty	.75	1	2	2	100	0.01859949	0.03477878
BS	chatty	.75	1	2	2	150	0.01224830	0.02011882
BS	chatty	.75	2	1	1	100	0.01383465	0.02555118
BS	chatty	.75	2	1	2	100	0.01596037	0.03281200
BS	chatty	.75	2	1	2	150	0.01619309	0.03129762
BS	chatty	.75	2	2	1	100	0.01379217	0.02660748
BS	chatty	.75	2	2	2	100	0.01564946	0.03186266
BS	chatty	.75	2	2	2	150	0.01405599	0.02619661

APPENDIX B

FINAL SIMULATION ANALYSIS TABLES

Factor	mean(mean)	mean(stdev)
$\lambda \times T = 100$	.01559	.03010
$\lambda \times T = 200$	.01525	.02929
$\lambda \times T = 300$	.01529	.02992
$\rho = \psi = 1$	.01562	.03081
$\rho = \psi = 2$	.01513	.02873
$\gamma = \eta = 1$	.01534	.02959
$\gamma = \eta = 2$	.01542	.02994
$\delta = 0.25$	.01484	.02775
$\delta = 0.50$	.01509	.02985
$\delta = 0.75$	.01621	.03170

Table B.1: Factor Summaries for Ω_{FFA}.

Factor	mean(mean)	mean(stdev)
$\lambda \times T = 100$	.01564	.03155
$\lambda \times T = 200$	.01608	.03041
$\lambda \times T = 300$	.01564	.03116
$\rho = \psi = 1$	.01575	.03172
$\rho = \psi = 2$	.01582	.03036
$\gamma = \eta = 1$	.01560	.03056
$\gamma = \eta = 2$	.01597	.03152
$\delta = 0.25$	.01593	.03220
$\delta = 0.50$	.01587	.03087
$\delta = 0.75$	.01556	.03005

Table B.2: Factor Summaries for $\Omega_{CL}(unif)$.

Factor	mean(mean)	mean(stdev)
$\lambda \times T = 100$	.01568	.02960
$\lambda \times T = 200$	.01549	.02898
$\lambda \times T = 300$	.01507	.02890
$\rho = \psi = 1$	.01533	.02917
$\rho = \psi = 2$	.01549	.02914
$\gamma = \eta = 1$	.01493	.02797
$\gamma = \eta = 2$	.01590	.03035
$\delta = 0.25$	.01575	.03020
$\delta = 0.50$	.01518	.02827
$\delta = 0.75$	.01530	.02900

Table B.3: Factor Summaries for $\Omega_{CL}(chatty)$.

Factor	mean(mean)	mean(stdev)
$\lambda \times T = 100$	.01495	.02800
$\lambda \times T = 200$	.01667	.01397
$\lambda \times T = 300$	.01533	.03061
$\rho = \psi = 1$	.01619	.03217
$\rho = \psi = 2$	.01524	.02822
$\gamma = \eta = 1$	.01575	.03019
$\gamma = \eta = 2$	.01569	.03020
$\delta = 0.25$	.01584	.03053
$\delta = 0.50$	.01481	.02733
$\delta = 0.75$	.01650	.03272

Table B.4: Factor Summaries for $\Omega_{BS}(unif)$.

Factor	mean(mean)	mean(stdev)
$\lambda \times T = 100$	.01600	.03157
$\lambda \times T = 200$	.01568	.03033
$\lambda \times T = 300$	.01494	.02907
$\rho = \psi = 1$	.01574	.03076
$\rho = \psi = 2$	.01534	.02989
$\gamma = \eta = 1$	.01512	.02964
$\gamma = \eta = 2$	.01595	.03100
$\delta = 0.25$	.01520	.03044
$\delta = 0.50$	.01603	.03116
$\delta = 0.75$	.01539	.02937

Table B.5: Factor Summaries for $\Omega_{BS}(chatty)$.

APPENDIX C

R-CODE

```
##################
## Simulation 1 ##
##################

sim1 <- function(lambda,Omega,m1,T,Zeta){
  tau <- rpois(1, lambda*T)
  t <- sort(runif(tau,0,T))
  w <- rbinom(tau,1,m1)+1
  z <- integer(length(w))
  for(i in 1:length(w)){
    if(w[i]==1)
      Zeta[2] <- Omega[1,2]*Zeta[1]+(1-Omega[1,2])*Zeta[2]
    if(w[i]==2)
      Zeta[1] <- Omega[2,1]*Zeta[2]+(1-Omega[2,1])*Zeta[1]
    z[i] <- Zeta[w[i]]
  }
  return(list(t=t, w=w, z=z))
}

###################
## Testing Sim 1 ##
###################

lambda=2
m1 <- .5
T <- 1
```

```
Zeta <- c(0,1)
Omega <- matrix(c(0,.75,.5,0),2,2,byrow=TRUE)

set.seed(157892)
sim1(lambda,Omega,m1,T,Zeta)
$t
[1] 0.02170549 0.30825124 0.63838089 0.95156962

$w
[1] 2 1 2 1

$z
[1] 1.0000 0.5000 0.6250 0.5625


##############################
## Generalized Simulation 2 ##
##############################

sim2 <- function(lambda, Omega, m, T, d, rho, psi){
  Zeta <- rbeta(d,rho,psi)
  tau <- rpois(1, lambda*T)
  t <- sort(runif(tau,0,T))
  w <- sample(1:d,tau,prob=m, replace=TRUE)
  z <- double(length(w))
  for (i in 1:tau){
```

```
    for (j in 1:d){
      if (j!=w[i]){
        Zeta[j] <- Omega[w[i],j]*Zeta[w[i]]+(1-Omega[w[i],j])*Zeta[j]
  }}
  z[i] <- Zeta[w[i]]
  }

  return(list(t=t, w=w, z=z, Zeta=Zeta))
}

###############################
## Testing Generalized Sim 2 ##
###############################

lambda <- 2
T <- 1
d <- 6
rho <- 1
psi <- 2

Omega <- matrix(.5,d,d)
diag(Omega) <- 0
m <- rep(1/d,d)

set.seed(157894)
sim2(lambda, Omega, m, T, d, rho, psi)
```

```
$t
[1] 0.2109341 0.5036478 0.8588514


$w
[1] 2 5 4


$z
[1] 0.03570841 0.21695110 0.20812536


$Zeta
[1] 0.2359339 0.1672276 0.2029616 0.2081254 0.2125382 0.1733964


#################
## Final Sim 2 ##
#################


sim2 <- function(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F) {
 zeta <- rbeta(d,rho,psi)
 tau <- max(rpois(1,lambda*T),1)
 t <- sort(runif(tau,0,T))
 w <- sample(1:d,tau,prob=m,replace=T)
 z <- matrix(0,tau,1)
 u <- rbeta(tau,gamma,eta)
 for(i in 1:tau) {
   for(j in 1:d)  {
```

```
      if(verbose) {
        print(i)
        print(j)
        print(w)
        print(z)
        print(Omega)
        print(zeta)
        print(u)
        print(Delta)
      }

      if(j!=w[i]) zeta[j] <- Omega[w[i],j]*zeta[w[i]] +
          (1-Delta[w[i]])*(1-Omega[w[i],j])*zeta[j]+
          Delta[w[i]]*(1-Omega[w[i],j])*u[i]
    }
    z[i] <- zeta[w[i]]
  }
  return(list(t=t,w=w,z=z,d=d,T=T))
}

###################
## Making Omegas ##
###################

FFA <- matrix(0.5,d,d)
diag(FFA) <- 0
```

```
CL <- matrix(0,d,d)
CL[1,] <- .5
diag(CL) <- 0

A <- matrix(.25,d/2,d/2)
A[1,] <- .75
A[,1] <- 0
diag(A) <- 0
B <- matrix(.1,d/2,d/2)
B[1,] <- .25
B[,1] <- 0
C <- matrix(.1,d/2,d/2)
C[,1] <- 0
D <- A
AC <- rbind(A,C)
BD <- rbind(B,D)
BS <- cbind(AC,BD)

#######################
## Plotting Function ##
#######################

plot.opinions <- function(x,verbose=F) {
  xl <- c(0,x$T)
  yl <- c(0,1)
```

```
  w.table <- table(factor(x$w,levels=1:d))
  plot(xl,yl,type="n",xlim=xl,ylim=yl,xlab="t",ylab="z")
  for(i in 1:x$d) {
    if(w.table[i]>0) {
      ts <- c(x$t[x$w==i])
      os <- c(x$z[x$w==i])
      if(verbose) {
        print(i)
        print(ts)
        print(os)
      }
      lines(ts,os,col=i)
    }
  }
}

#######################
## Pilot Simulations ##
#######################
### Pilot Sim 1 ###
lambda <- 1
T <- 10
d <- 10
rho <- 2
psi <- 2
gamma <- 2
```

```
eta <- 2
Delta <- rep(0.25,d)

Omega <- matrix(0.5,d,d)
diag(Omega) <- 0
m <- rep(1/d,d)

x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
x
plot.opinions(x,verbose=F)

### Pilot Sim 2 ###
lambda <- 1
T <- 10
d <- 10
rho <- 2
psi <- 2
gamma <- 2
eta <- 2
Delta <- rep(0.75,d)

Omega <- matrix(0.5,d,d)
diag(Omega) <- 0
m <- rep(1/d,d)

x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
```

```
x
plot.opinions(x,verbose=F)

### Pilot Sim 3 ###
lambda <- 1
T <- 100
d <- 10
rho <- 2
psi <- 2
gamma <- 2
eta <- 2
Delta <- rep(0.25,d)

Omega <- matrix(0.5,d,d)
diag(Omega) <- 0
m <- rep(1/d,d)

x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
x
plot.opinions(x,verbose=F)

### Pilot Sim 4 ###
lambda <- 1
T <- 100
d <- 10
rho <- 2
```

```
psi <- 2
gamma <- 2
eta <- 2
Delta <- rep(0.75,d)

Omega <- matrix(0.5,d,d)
diag(Omega) <- 0
m <- rep(1/d,d)

x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
x
plot.opinions(x,verbose=F)

### Pilot Sim 5 ###
lambda <- 1
T <- 10
d <- 10
rho <- 2
psi <- 2
gamma <- 2
eta <- 2
Delta <- rep(0.25,d)

Omega <- matrix(0,d,d)
Omega[,1]<-.5
diag(Omega) <- 0
```

```
m <- rep(1/d,d)

x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
x
plot.opinions(x,verbose=F)

### Pilot Sim 6 ###
lambda <- 1
T <- 10
d <- 10
rho <- 2
psi <- 2
gamma <- 2
eta <- 2
Delta <- rep(0.75,d)

Omega <- matrix(0,d,d)
Omega[,1]<-.5
diag(Omega) <- 0
m <- rep(1/d,d)

x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
x
plot.opinions(x,verbose=F)

### Pilot Sim 7 ###
```

```
lambda <- 1
T <- 100
d <- 10
rho <- 2
psi <- 2
gamma <- 2
eta <- 2
Delta <- rep(0.25,d)

Omega <- matrix(0,d,d)
Omega[,1]<-.5
diag(Omega) <- 0
m <- rep(1/d,d)

x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
x
plot.opinions(x,verbose=F)

### Pilot Sim 8 ###
lambda <- 1
T <- 100
d <- 10
rho <- 2
psi <- 2
gamma <- 2
eta <- 2
```

```
Delta <- rep(0.75,d)

Omega <- matrix(0,d,d)
Omega[,1]<-.5
diag(Omega) <- 0
m <- rep(1/d,d)

x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
x
plot.opinions(x,verbose=F)

### Pilot Sim 9 ###
lambda <- 1
T <- 10
d <- 10
rho <- 2
psi <- 2
gamma <- 2
eta <- 2
Delta <- rep(0.25,d)

A <- matrix(.25,d/2,d/2)
A[,1]<-.75
A[1,]<-0
diag(A)<-0
B<-matrix(.1,d/2,d/2)
```

```
B[1,]<-0
C<-matrix(.1,d/2,d/2)
C[,1]<-.25
C[1,]<-0
D<-A
AC <- rbind(A,C)
BD <- rbind(B,D)
Omega <- cbind(AC,BD)
m <- rep(1/d,d)

x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
x
plot.opinions(x,verbose=F)

### Pilot Sim 10 ###
lambda <- 1
T <- 10
d <- 10
rho <- 2
psi <- 2
gamma <- 2
eta <- 2
Delta <- rep(0.75,d)

A <- matrix(.25,d/2,d/2)
A[,1]<-.75
```

```
A[1,]<-0
diag(A)<-0
B<-matrix(.1,d/2,d/2)
B[1,]<-0
C<-matrix(.1,d/2,d/2)
C[,1]<-.25
C[1,]<-0
D<-A
AC <- rbind(A,C)
BD <- rbind(B,D)
Omega <- cbind(AC,BD)
m <- rep(1/d,d)

x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
x
plot.opinions(x,verbose=F)

### Pilot Sim 11 ###
lambda <- 1
T <- 100
d <- 10
rho <- 2
psi <- 2
gamma <- 2
eta <- 2
Delta <- rep(0.25,d)
```

```
A <- matrix(.25,d/2,d/2)
A[,1]<-.75
A[1,]<-0
diag(A)<-0
B<-matrix(.1,d/2,d/2)
B[1,]<-0
C<-matrix(.1,d/2,d/2)
C[,1]<-.25
C[1,]<-0
D<-A
AC <- rbind(A,C)
BD <- rbind(B,D)
Omega <- cbind(AC,BD)
m <- rep(1/d,d)

x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
x
plot.opinions(x,verbose=F)

### Pilot Sim 12 ###
lambda <- 1
T <- 100
d <- 10
rho <- 2
psi <- 2
```

```
gamma <- 2
eta <- 2
Delta <- rep(0.75,d)

A <- matrix(.25,d/2,d/2)
A[,1]<-.75
A[1,]<-0
diag(A)<-0
B<-matrix(.1,d/2,d/2)
B[1,]<-0
C<-matrix(.1,d/2,d/2)
C[,1]<-.25
C[1,]<-0
D<-A
AC <- rbind(A,C)
BD <- rbind(B,D)
Omega <- cbind(AC,BD)
m <- rep(1/d,d)

x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
x
plot.opinions(x,verbose=F)

###########################
## Secondary Simulations ##
###########################
```

```
###Simulation 1###
lambda <- 1
T <- 100
d <- 10
rho <- 2
psi <- 2
gamma <- 2
eta <- 2
Delta <- rep(0,d)

Omega <- matrix(0.5,d,d)
diag(Omega) <- 0
m <- rep(1/d,d)

x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
x
plot.opinions(x,verbose=F)

###Simulation 2###
lambda <- 1
T <- 1000
d <- 10
rho <- 2
psi <- 2
gamma <- 2
```

```
eta <- 2
Delta <- rep(0,d)

Omega <- matrix(0.5,d,d)
diag(Omega) <- 0
m <- rep(1/d,d)

x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
x
plot.opinions(x,verbose=F)

###Simulation 3###
lambda <- 1
T <- 100
d <- 10
rho <- 2
psi <- 2
gamma <- 2
eta <- 2
Delta <- rep(0,d)

Omega <- matrix(0,d,d)
Omega[1,]<-.5
diag(Omega) <- 0
m <- rep(1/d,d)
```

```
x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
x
plot.opinions(x,verbose=F)

###Simulation 4###
lambda <- 1
T <- 1000
d <- 10
rho <- 2
psi <- 2
gamma <- 2
eta <- 2
Delta <- rep(0,d)

Omega <- matrix(0,d,d)
Omega[,1]<-.5
diag(Omega) <- 0
m <- rep(1/d,d)

x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
x
plot.opinions(x,verbose=F)

###Simulation 5###
lambda <- 1
T <- 100
```

```
d <- 10
rho <- 2
psi <- 2
gamma <- 2
eta <- 2
Delta <- rep(0,d)

A <- matrix(.25,d/2,d/2)
A[1,]<-.75
A[,1]<-0
diag(A)<-0
B <- matrix(.1,d/2,d/2)
B[1,]<-.25
B[,1]<-0
C<-matrix(.1,d/2,d/2)
C[,1]<-0
D<-A
AC <- rbind(A,C)
BD <- rbind(B,D)
Omega <- cbind(AC,BD)
m <- rep(1/d,d)

x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
x
plot.opinions(x,verbose=F)
```

```
###Simulation 6###
lambda <- 1
T <- 1000
d <- 10
rho <- 2
psi <- 2
gamma <- 2
eta <- 2
Delta <- rep(0,d)

A <- matrix(.25,d/2,d/2)
A[1,]<-.75
A[,1]<-0
diag(A)<-0
B <- matrix(.1,d/2,d/2)
B[1,]<-.25
B[,1]<-0
C<-matrix(.1,d/2,d/2)
C[,1]<-0
D<-A
AC <- rbind(A,C)
BD <- rbind(B,D)
Omega <- cbind(AC,BD)
m <- rep(1/d,d)

x <- sim2(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, verbose=F)
```

```
x
plot.opinions(x,verbose=F)

#########################
## Simulation Function ##
#########################

simulation <- function(lambda, Omega, m, Delta, T, d, rho, psi, gamma, eta, b=1000) {
  a <- vector(mode="double",length=b)
  for(i in 1:b) a[i] <- mad(sim2(lambda,Omega,m,Delta,T,d,rho,psi,gamma,eta)$zeta)
  return(a)
}

#######################
## Final Simulations ##
#######################
FFA.unif.25.1.1.100
<- simulation(lambda=1,Omega=FFA,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
FFA.unif.25.1.1.200
<- simulation(lambda=2,Omega=FFA,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
FFA.unif.25.1.1.300
<- simulation(lambda=2,Omega=FFA,m,delta,T=150,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
FFA.unif.25.1.2.100
<- simulation(lambda=1,Omega=FFA,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
FFA.unif.25.1.2.200
<- simulation(lambda=2,Omega=FFA,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
```

```
FFA.unif.25.1.2.300
<- simulation(lambda=2,Omega=FFA,m,delta,T=150,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
FFA.unif.25.2.2.100
<- simulation(lambda=1,Omega=FFA,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
FFA.unif.25.2.2.200
<- simulation(lambda=2,Omega=FFA,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
FFA.unif.25.2.2.300
<- simulation(lambda=2,Omega=FFA,m,delta,T=150,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
FFA.unif.25.2.1.100
<- simulation(lambda=1,Omega=FFA,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
FFA.unif.25.2.1.200
<- simulation(lambda=2,Omega=FFA,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
FFA.unif.25.2.1.300
<- simulation(lambda=2,Omega=FFA,m,delta,T=150,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
delta <- rep(.5,d)
FFA.unif.50.1.1.100
<- simulation(lambda=1,Omega=FFA,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
FFA.unif.50.1.1.200
<- simulation(lambda=2,Omega=FFA,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
FFA.unif.50.1.1.300
<- simulation(lambda=2,Omega=FFA,m,delta,T=150,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
FFA.unif.50.1.2.100
<- simulation(lambda=1,Omega=FFA,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
FFA.unif.50.1.2.200
<- simulation(lambda=2,Omega=FFA,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
FFA.unif.50.1.2.300
```

```
<- simulation(lambda=2,Omega=FFA,m,delta,T=150,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
FFA.unif.50.2.2.100
<- simulation(lambda=1,Omega=FFA,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
FFA.unif.50.2.2.200
<- simulation(lambda=2,Omega=FFA,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
FFA.unif.50.2.2.300
<- simulation(lambda=2,Omega=FFA,m,delta,T=150,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
FFA.unif.50.2.1.100
<- simulation(lambda=1,Omega=FFA,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
FFA.unif.50.2.1.200
<- simulation(lambda=2,Omega=FFA,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
FFA.unif.50.2.1.300
<- simulation(lambda=2,Omega=FFA,m,delta,T=150,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
delta <- rep(.75,d)
FFA.unif.75.1.1.100
<- simulation(lambda=1,Omega=FFA,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
FFA.unif.75.1.1.200
<- simulation(lambda=2,Omega=FFA,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
FFA.unif.75.1.1.300
<- simulation(lambda=2,Omega=FFA,m,delta,T=150,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
FFA.unif.75.1.2.100
<- simulation(lambda=1,Omega=FFA,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
FFA.unif.75.1.2.200
<- simulation(lambda=2,Omega=FFA,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
FFA.unif.75.1.2.300
<- simulation(lambda=2,Omega=FFA,m,delta,T=150,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
```

```
FFA.unif.75.2.2.100
<- simulation(lambda=1,Omega=FFA,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
FFA.unif.75.2.2.200
<- simulation(lambda=2,Omega=FFA,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
FFA.unif.75.2.2.300
<- simulation(lambda=2,Omega=FFA,m,delta,T=150,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
FFA.unif.75.2.1.100
<- simulation(lambda=1,Omega=FFA,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
FFA.unif.75.2.1.200
<- simulation(lambda=2,Omega=FFA,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
FFA.unif.75.2.1.300
<- simulation(lambda=2,Omega=FFA,m,delta,T=150,d,rho=2,psi=2,gamma=1,eta=1,b=1000)

delta <- rep(.25,d)
CL.unif.25.1.1.100
<- simulation(lambda=1,Omega=CL,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.unif.25.1.1.200
<- simulation(lambda=2,Omega=CL,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.unif.25.1.1.300
<- simulation(lambda=2,Omega=CL,m,delta,T=150,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.unif.25.1.2.100
<- simulation(lambda=1,Omega=CL,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
CL.unif.25.1.2.200
<- simulation(lambda=2,Omega=CL,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
CL.unif.25.1.2.300
<- simulation(lambda=2,Omega=CL,m,delta,T=150,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
```

```
CL.unif.25.2.2.100
<- simulation(lambda=1,Omega=CL,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
CL.unif.25.2.2.200
<- simulation(lambda=2,Omega=CL,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
CL.unif.25.2.2.300
<- simulation(lambda=2,Omega=CL,m,delta,T=150,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
CL.unif.25.2.1.100
<- simulation(lambda=1,Omega=CL,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
CL.unif.25.2.1.200
<- simulation(lambda=2,Omega=CL,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
CL.unif.25.2.1.300
<- simulation(lambda=2,Omega=CL,m,delta,T=150,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
delta <- rep(.5,d)
CL.unif.50.1.1.100
<- simulation(lambda=1,Omega=CL,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.unif.50.1.1.200
<- simulation(lambda=2,Omega=CL,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.unif.50.1.1.300
<- simulation(lambda=2,Omega=CL,m,delta,T=150,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.unif.50.1.2.100
<- simulation(lambda=1,Omega=CL,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
CL.unif.50.1.2.200
<- simulation(lambda=2,Omega=CL,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
CL.unif.50.1.2.300
<- simulation(lambda=2,Omega=CL,m,delta,T=150,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
CL.unif.50.2.2.100
```

```
<- simulation(lambda=1,Omega=CL,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
CL.unif.50.2.2.200
<- simulation(lambda=2,Omega=CL,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
CL.unif.50.2.2.300
<- simulation(lambda=2,Omega=CL,m,delta,T=150,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
CL.unif.50.2.1.100
<- simulation(lambda=1,Omega=CL,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
CL.unif.50.2.1.200
<- simulation(lambda=2,Omega=CL,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
CL.unif.50.2.1.300
<- simulation(lambda=2,Omega=CL,m,delta,T=150,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
delta <- rep(.75,d)
CL.unif.75.1.1.100
<- simulation(lambda=1,Omega=CL,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.unif.75.1.1.200
<- simulation(lambda=2,Omega=CL,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.unif.75.1.1.300
<- simulation(lambda=2,Omega=CL,m,delta,T=150,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.unif.75.1.2.100
<- simulation(lambda=1,Omega=CL,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
CL.unif.75.1.2.200
<- simulation(lambda=2,Omega=CL,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
CL.unif.75.1.2.300
<- simulation(lambda=2,Omega=CL,m,delta,T=150,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
CL.unif.75.2.2.100
<- simulation(lambda=1,Omega=CL,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
```

```
CL.unif.75.2.2.200
<- simulation(lambda=2,Omega=CL,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
CL.unif.75.2.2.300
<- simulation(lambda=2,Omega=CL,m,delta,T=150,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
CL.unif.75.2.1.100
<- simulation(lambda=1,Omega=CL,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
CL.unif.75.2.1.200
<- simulation(lambda=2,Omega=CL,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
CL.unif.75.2.1.300
<- simulation(lambda=2,Omega=CL,m,delta,T=150,d,rho=2,psi=2,gamma=1,eta=1,b=1000)

m2 <- c(.5,1/18,1/18,1/18,1/18,1/18,1/18,1/18,1/18,1/18)
delta <- rep(.25,d,d)
CL.chatty.25.1.1.100
<- simulation(lambda=1,Omega=CL,m=m2,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.chatty.25.1.1.200
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.chatty.25.1.1.300
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=150,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.chatty.25.1.2.100
<- simulation(lambda=1,Omega=CL,m=m2,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
CL.chatty.25.1.2.200
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
CL.chatty.25.1.2.300
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=150,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
CL.chatty.25.2.2.100
```

```
<- simulation(lambda=1,Omega=CL,m=m2,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
CL.chatty.25.2.2.200
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
CL.chatty.25.2.2.300
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=150,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
CL.chatty.25.2.1.100
<- simulation(lambda=1,Omega=CL,m=m2,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
CL.chatty.25.2.1.200
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
CL.chatty.25.2.1.300
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=150,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
delta <- rep(.5,d)
CL.chatty.50.1.1.100
<- simulation(lambda=1,Omega=CL,m=m2,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.chatty.50.1.1.200
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.chatty.50.1.1.300
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=150,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.chatty.50.1.2.100
<- simulation(lambda=1,Omega=CL,m=m2,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
CL.chatty.50.1.2.200
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
CL.chatty.50.1.2.300
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=150,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
CL.chatty.50.2.2.100
<- simulation(lambda=1,Omega=CL,m=m2,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
```

```
CL.chatty.50.2.2.200
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
CL.chatty.50.2.2.300
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=150,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
CL.chatty.50.2.1.100
<- simulation(lambda=1,Omega=CL,m=m2,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
CL.chatty.50.2.1.200
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
CL.chatty.50.2.1.300
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=150,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
delta <- rep(.75,d)
CL.chatty.75.1.1.100
<- simulation(lambda=1,Omega=CL,m=m2,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.chatty.75.1.1.200
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.chatty.75.1.1.300
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=150,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
CL.chatty.75.1.2.100
<- simulation(lambda=1,Omega=CL,m=m2,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
CL.chatty.75.1.2.200
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
CL.chatty.75.1.2.300
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=150,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
CL.chatty.75.2.2.100
<- simulation(lambda=1,Omega=CL,m=m2,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
CL.chatty.75.2.2.200
```

```
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
CL.chatty.75.2.2.300
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=150,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
CL.chatty.75.2.1.100
<- simulation(lambda=1,Omega=CL,m=m2,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
CL.chatty.75.2.1.200
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
CL.chatty.75.2.1.300
<- simulation(lambda=2,Omega=CL,m=m2,delta,T=150,d,rho=2,psi=2,gamma=1,eta=1,b=1000)

delta <- rep(.25,d)
BS.unif.25.1.1.100
<- simulation(lambda=1,Omega=BS,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.unif.25.1.1.200
<- simulation(lambda=2,Omega=BS,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.unif.25.1.1.300
<- simulation(lambda=2,Omega=BS,m,delta,T=150,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.unif.25.1.2.100
<- simulation(lambda=1,Omega=BS,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.unif.25.1.2.200
<- simulation(lambda=2,Omega=BS,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.unif.25.1.2.300
<- simulation(lambda=2,Omega=BS,m,delta,T=150,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.unif.25.2.2.100
<- simulation(lambda=1,Omega=BS,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
BS.unif.25.2.2.200
```

```
<- simulation(lambda=2,Omega=BS,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
BS.unif.25.2.2.300
<- simulation(lambda=2,Omega=BS,m,delta,T=150,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
BS.unif.25.2.1.100
<- simulation(lambda=1,Omega=BS,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
BS.unif.25.2.1.200
<- simulation(lambda=2,Omega=BS,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
BS.unif.25.2.1.300
<- simulation(lambda=2,Omega=BS,m,delta,T=150,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
delta <- rep(.5,d)
BS.unif.50.1.1.100
<- simulation(lambda=1,Omega=BS,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.unif.50.1.1.200
<- simulation(lambda=2,Omega=BS,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.unif.50.1.1.300
<- simulation(lambda=2,Omega=BS,m,delta,T=150,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.unif.50.1.2.100
<- simulation(lambda=1,Omega=BS,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.unif.50.1.2.200
<- simulation(lambda=2,Omega=BS,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.unif.50.1.2.300
<- simulation(lambda=2,Omega=BS,m,delta,T=150,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.unif.50.2.2.100
<- simulation(lambda=1,Omega=BS,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
BS.unif.50.2.2.200
<- simulation(lambda=2,Omega=BS,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
```

```
BS.unif.50.2.2.300
<- simulation(lambda=2,Omega=BS,m,delta,T=150,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
BS.unif.50.2.1.100
<- simulation(lambda=1,Omega=BS,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
BS.unif.50.2.1.200
<- simulation(lambda=2,Omega=BS,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
BS.unif.50.2.1.300
<- simulation(lambda=2,Omega=BS,m,delta,T=150,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
delta <- rep(.75,d)
BS.unif.75.1.1.100
<- simulation(lambda=1,Omega=BS,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.unif.75.1.1.200
<- simulation(lambda=2,Omega=BS,m,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.unif.75.1.1.300
<- simulation(lambda=2,Omega=BS,m,delta,T=150,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.unif.75.1.2.100
<- simulation(lambda=1,Omega=BS,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.unif.75.1.2.200
<- simulation(lambda=2,Omega=BS,m,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.unif.75.1.2.300
<- simulation(lambda=2,Omega=BS,m,delta,T=150,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.unif.75.2.2.100
<- simulation(lambda=1,Omega=BS,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
BS.unif.75.2.2.200
<- simulation(lambda=2,Omega=BS,m,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
BS.unif.75.2.2.300
```

```
<- simulation(lambda=2,Omega=BS,m,delta,T=150,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
BS.unif.75.2.1.100
<- simulation(lambda=1,Omega=BS,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
BS.unif.75.2.1.200
<- simulation(lambda=2,Omega=BS,m,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
BS.unif.75.2.1.300
<- simulation(lambda=2,Omega=BS,m,delta,T=150,d,rho=2,psi=2,gamma=1,eta=1,b=1000)

m3 <- c(.25,.0625,.0625,.0625,.0625,.25,.0625,.0625,.0625,.0625)
delta <- rep(.25,d)
BS.chatty.25.1.1.100
<- simulation(lambda=1,Omega=BS,m=m3,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.chatty.25.1.1.200
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.chatty.25.1.1.300
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=150,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.chatty.25.1.2.100
<- simulation(lambda=1,Omega=BS,m=m3,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.chatty.25.1.2.200
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.chatty.25.1.2.300
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=150,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.chatty.25.2.2.100
<- simulation(lambda=1,Omega=BS,m=m3,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
BS.chatty.25.2.2.200
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
```

```
BS.chatty.25.2.2.300
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=150,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
BS.chatty.25.2.1.100
<- simulation(lambda=1,Omega=BS,m=m3,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
BS.chatty.25.2.1.200
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
BS.chatty.25.2.1.300
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=150,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
delta <- rep(.5,d)
BS.chatty.50.1.1.100
<- simulation(lambda=1,Omega=BS,m=m3,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.chatty.50.1.1.200
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.chatty.50.1.1.300
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=150,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.chatty.50.1.2.100
<- simulation(lambda=1,Omega=BS,m=m3,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.chatty.50.1.2.200
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.chatty.50.1.2.300
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=150,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.chatty.50.2.2.100
<- simulation(lambda=1,Omega=BS,m=m3,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
BS.chatty.50.2.2.200
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
BS.chatty.50.2.2.300
```

```
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=150,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
BS.chatty.50.2.1.100
<- simulation(lambda=1,Omega=BS,m=m3,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
BS.chatty.50.2.1.200
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
BS.chatty.50.2.1.300
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=150,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
delta <- rep(.75,d)
BS.chatty.75.1.1.100
<- simulation(lambda=1,Omega=BS,m=m3,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.chatty.75.1.1.200
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=100,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.chatty.75.1.1.300
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=150,d,rho=1,psi=1,gamma=1,eta=1,b=1000)
BS.chatty.75.1.2.100
<- simulation(lambda=1,Omega=BS,m=m3,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.chatty.75.1.2.200
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=100,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.chatty.75.1.2.300
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=150,d,rho=1,psi=1,gamma=2,eta=2,b=1000)
BS.chatty.75.2.2.100
<- simulation(lambda=1,Omega=BS,m=m3,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
BS.chatty.75.2.2.200
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=100,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
BS.chatty.75.2.2.300
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=150,d,rho=2,psi=2,gamma=2,eta=2,b=1000)
```

```
BS.chatty.75.2.1.100
<- simulation(lambda=1,Omega=BS,m=m3,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
BS.chatty.75.2.1.200
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=100,d,rho=2,psi=2,gamma=1,eta=1,b=1000)
BS.chatty.75.2.1.300
<- simulation(lambda=2,Omega=BS,m=m3,delta,T=150,d,rho=2,psi=2,gamma=1,eta=1,b=1000)


########################
## summary statistics ##
########################

summary(alldata.table[c(8,9)])
      mean              stdev
 Min.   :0.01049   Min.   :0.01680
 1st Qu.:0.01426   1st Qu.:0.02670
 Median :0.01562   Median :0.03000
 Mean   :0.01557   Mean   :0.03010
 3rd Qu.:0.01677   3rd Qu.:0.03362
 Max.   :0.02083   Max.   :0.04570
hist(alldata.table$mean)
hist(alldata.table$stdev)

## comparing factors within each omega##
library(dplyr)
lambt100 <- alldata %>% filter(lambdaT==100)
lambt200 <- alldata %>% filter(lambdaT==200)
```

```
lambt300 <- alldata %>% filter(lambdaT==300)
rhopsi1 <- alldata %>% filter(rho.psi==1)
rhopsi2 <- alldata %>% filter(rho.psi==2)
gamet1 <- alldata %>% filter(gamma.eta==1)
gamet2 <- alldata %>% filter(gamma.eta==2)
delt25 <- alldata %>% filter(delta==25)
delt50 <- alldata %>% filter(delta==50)
delt75 <- alldata %>% filter(delta==75)


#FFAlambt100
summary(lambt100[1:12,c(1008,1009)])
     mean              stdev
Min.   :0.01224   Min.   :0.02123
1st Qu.:0.01415   1st Qu.:0.02648
Median :0.01541   Median :0.02889
Mean   :0.01559   Mean   :0.03010
3rd Qu.:0.01700   3rd Qu.:0.03224
Max.   :0.02083   Max.   :0.04464
#FFAlambt200
summary(lambt200[1:12,c(1008,1009)])
     mean              stdev
Min.   :0.01323   Min.   :0.02517
1st Qu.:0.01415   1st Qu.:0.02701
Median :0.01489   Median :0.02769
Mean   :0.01525   Mean   :0.02929
```

```
3rd Qu.:0.01630   3rd Qu.:0.03317
Max.   :0.01770   Max.   :0.03444
#FFAlambt300
summary(lambt300[1:12,c(1008,1009)])
     mean              stdev
Min.   :0.01049   Min.   :0.01680
1st Qu.:0.01430   1st Qu.:0.02929
Median :0.01575   Median :0.02985
Mean   :0.01529   Mean   :0.02992
3rd Qu.:0.01641   3rd Qu.:0.03245
Max.   :0.01812   Max.   :0.03626
#FFArhopsi1
summary(rhopsi1[1:18,c(1008,1009)])
     mean              stdev
Min.   :0.01323   Min.   :0.02517
1st Qu.:0.01436   1st Qu.:0.02768
Median :0.01506   Median :0.02966
Mean   :0.01562   Mean   :0.03081
3rd Qu.:0.01645   3rd Qu.:0.03241
Max.   :0.02083   Max.   :0.04464
#FFArhopsi2
summary(rhopsi2[1:18,c(1008,1009)])
     mean              stdev
Min.   :0.01049   Min.   :0.01680
1st Qu.:0.01373   1st Qu.:0.02687
Median :0.01566   Median :0.02885
```

```
Mean   :0.01513   Mean   :0.02873
3rd Qu.:0.01644   3rd Qu.:0.03217
Max.   :0.01812   Max.   :0.03626
#FFAgamet1
summary(gamet1[1:18,c(1008,1009)])
     mean              stdev
Min.   :0.01276   Min.   :0.02431
1st Qu.:0.01377   1st Qu.:0.02678
Median :0.01516   Median :0.02834
Mean   :0.01534   Mean   :0.02959
3rd Qu.:0.01638   3rd Qu.:0.03176
Max.   :0.02083   Max.   :0.04464
#FFAgamet2
summary(gamet2[1:18,c(1008,1009)])
     mean              stdev
Min.   :0.01049   Min.   :0.01680
1st Qu.:0.01441   1st Qu.:0.02786
Median :0.01549   Median :0.03023
Mean   :0.01542   Mean   :0.02994
3rd Qu.:0.01669   3rd Qu.:0.03324
Max.   :0.01812   Max.   :0.03757
#FFAdelt25
summary(delt25[1:12,c(1008,1009)])
     mean              stdev
Min.   :0.01224   Min.   :0.02123
1st Qu.:0.01420   1st Qu.:0.02674
```

```
Median :0.01502   Median :0.02769
Mean   :0.01484   Mean   :0.02775
3rd Qu.:0.01581   3rd Qu.:0.02976
Max.   :0.01648   Max.   :0.03238
#FFAdelt50
summary(delt50[1:12,c(1008,1009)])
     mean              stdev
Min.   :0.01049   Min.   :0.01680
1st Qu.:0.01360   1st Qu.:0.02841
Median :0.01571   Median :0.03148
Mean   :0.01509   Mean   :0.02985
3rd Qu.:0.01658   3rd Qu.:0.03276
Max.   :0.01770   Max.   :0.03469
#FFAdelt75
summary(delt75[1:12,c(1008,1009)])
     mean              stdev
Min.   :0.01372   Min.   :0.02572
1st Qu.:0.01467   1st Qu.:0.02781
Median :0.01572   Median :0.02950
Mean   :0.01621   Mean   :0.03170
3rd Qu.:0.01756   3rd Qu.:0.03422
Max.   :0.02083   Max.   :0.04464


#CLuniflambt100
summary(lambt100[13:24,c(1008,1009)])
```

```
     mean             stdev
Min.   :0.01306   Min.   :0.02214
1st Qu.:0.01458   1st Qu.:0.02972
Median :0.01567   Median :0.03192
Mean   :0.01564   Mean   :0.03155
3rd Qu.:0.01642   3rd Qu.:0.03366
Max.   :0.01920   Max.   :0.04142
#CLuniflambt200
summary(lambt200[13:24,c(1008,1009)])
     mean             stdev
Min.   :0.01383   Min.   :0.02222
1st Qu.:0.01422   1st Qu.:0.02534
Median :0.01594   Median :0.03011
Mean   :0.01608   Mean   :0.03041
3rd Qu.:0.01799   3rd Qu.:0.03653
Max.   :0.01859   Max.   :0.03768
#CLuniflambt300
summary(lambt300[13:24,c(1008,1009)])
     mean             stdev
Min.   :0.01373   Min.   :0.02571
1st Qu.:0.01444   1st Qu.:0.02727
Median :0.01477   Median :0.03132
Mean   :0.01564   Mean   :0.03116
3rd Qu.:0.01703   3rd Qu.:0.03486
Max.   :0.01882   Max.   :0.03666
#CLunifrhopsi1
```

```
summary(rhopsi1[19:36,c(1008,1009)])
     mean              stdev
Min.   :0.01306   Min.   :0.02297
1st Qu.:0.01430   1st Qu.:0.02848
Median :0.01598   Median :0.03187
Mean   :0.01575   Mean   :0.03172
3rd Qu.:0.01668   3rd Qu.:0.03603
Max.   :0.01920   Max.   :0.04142
#CLunifrhopsi2
summary(rhopsi2[19:36,c(1008,1009)])
     mean              stdev
Min.   :0.01356   Min.   :0.02214
1st Qu.:0.01433   1st Qu.:0.02629
Median :0.01535   Median :0.03049
Mean   :0.01582   Mean   :0.03036
3rd Qu.:0.01756   3rd Qu.:0.03546
Max.   :0.01882   Max.   :0.03768
#CLunifgamet1
summary(gamet1[19:36,c(1008,1009)])
     mean              stdev
Min.   :0.01356   Min.   :0.02214
1st Qu.:0.01417   1st Qu.:0.02735
Median :0.01467   Median :0.03080
Mean   :0.01560   Mean   :0.03056
3rd Qu.:0.01681   3rd Qu.:0.03227
Max.   :0.01920   Max.   :0.04142
```

```
#CLunifgamet2
summary(gamet2[19:36,c(1008,1009)])
     mean                 stdev
Min.   :0.01306   Min.   :0.02222
1st Qu.:0.01476   1st Qu.:0.02643
Median :0.01598   Median :0.03288
Mean   :0.01597   Mean   :0.03152
3rd Qu.:0.01778   3rd Qu.:0.03657
Max.   :0.01882   Max.   :0.03768
#CLunifdelt25
summary(delt25[13:24,c(1008,1009)])
     mean                 stdev
Min.   :0.01373   Min.   :0.02612
1st Qu.:0.01474   1st Qu.:0.02809
Median :0.01609   Median :0.03245
Mean   :0.01593   Mean   :0.03220
3rd Qu.:0.01691   3rd Qu.:0.03659
Max.   :0.01801   Max.   :0.03768
#CLunifdelt50
summary(delt50[13:24,c(1008,1009)])
     mean                 stdev
Min.   :0.01356   Min.   :0.02214
1st Qu.:0.01411   1st Qu.:0.02784
Median :0.01567   Median :0.03184
Mean   :0.01587   Mean   :0.03087
3rd Qu.:0.01772   3rd Qu.:0.03595
```

```
Max.   :0.01882   Max.   :0.03714
#CLunifdelt75
summary(delt75[13:24,c(1008,1009)])
     mean              stdev
Min.   :0.01306   Min.   :0.02297
1st Qu.:0.01422   1st Qu.:0.02534
Median :0.01470   Median :0.02957
Mean   :0.01556   Mean   :0.03005
3rd Qu.:0.01710   3rd Qu.:0.03316
Max.   :0.01920   Max.   :0.04142


#CLchatlambt100
summary(lambt100[25:36,c(1008,1009)])
     mean              stdev
Min.   :0.01296   Min.   :0.01974
1st Qu.:0.01459   1st Qu.:0.02615
Median :0.01547   Median :0.03142
Mean   :0.01568   Mean   :0.02960
3rd Qu.:0.01664   3rd Qu.:0.03323
Max.   :0.01883   Max.   :0.03467
#CLchatlambt200
summary(lambt200[25:36,c(1008,1009)])
     mean              stdev
Min.   :0.01318   Min.   :0.02310
1st Qu.:0.01485   1st Qu.:0.02695
```

```
Median :0.01543   Median :0.02911
Mean   :0.01549   Mean   :0.02898
3rd Qu.:0.01607   3rd Qu.:0.03136
Max.   :0.01830   Max.   :0.03416
#CLchatlambt300
summary(lambt300[25:36,c(1008,1009)])
     mean              stdev
Min.   :0.01147   Min.   :0.02073
1st Qu.:0.01393   1st Qu.:0.02455
Median :0.01586   Median :0.02900
Mean   :0.01507   Mean   :0.02890
3rd Qu.:0.01634   3rd Qu.:0.03350
Max.   :0.01705   Max.   :0.03526
#CLchatrhopsi1
summary(rhopsi1[37:54,c(1008,1009)])
     mean              stdev
Min.   :0.01296   Min.   :0.01974
1st Qu.:0.01426   1st Qu.:0.02659
Median :0.01530   Median :0.02900
Mean   :0.01533   Mean   :0.02917
3rd Qu.:0.01605   3rd Qu.:0.03319
Max.   :0.01883   Max.   :0.03526
#CLchatrhopsi2
summary(rhopsi2[37:54,c(1008,1009)])
     mean              stdev
Min.   :0.01147   Min.   :0.02073
```

```
1st Qu.:0.01475   1st Qu.:0.02546
Median :0.01590   Median :0.03066
Mean   :0.01549   Mean   :0.02914
3rd Qu.:0.01637   3rd Qu.:0.03296
Max.   :0.01830   Max.   :0.03467
#CLchatgamet1
summary(gamet1[37:54,c(1008,1009)])
     mean              stdev
Min.   :0.01296   Min.   :0.01974
1st Qu.:0.01368   1st Qu.:0.02366
Median :0.01542   Median :0.02829
Mean   :0.01493   Mean   :0.02797
3rd Qu.:0.01591   3rd Qu.:0.03235
Max.   :0.01639   Max.   :0.03473
#CLchatgamet2
summary(gamet2[37:54,c(1008,1009)])
     mean              stdev
Min.   :0.01147   Min.   :0.02073
1st Qu.:0.01471   1st Qu.:0.02762
Median :0.01617   Median :0.03081
Mean   :0.01590   Mean   :0.03035
3rd Qu.:0.01703   3rd Qu.:0.03318
Max.   :0.01883   Max.   :0.03526
#CLchatdelt25
summary(delt25[25:36,c(1008,1009)])
     mean              stdev
```

```
Min.   :0.01312   Min.   :0.02247
1st Qu.:0.01495   1st Qu.:0.02634
Median :0.01580   Median :0.03156
Mean   :0.01575   Mean   :0.03020
3rd Qu.:0.01645   3rd Qu.:0.03392
Max.   :0.01883   Max.   :0.03483
#CLchatdelt50
summary(delt50[25:36,c(1008,1009)])
     mean               stdev
Min.   :0.01147   Min.   :0.01974
1st Qu.:0.01396   1st Qu.:0.02545
Median :0.01572   Median :0.02863
Mean   :0.01518   Mean   :0.02827
3rd Qu.:0.01621   3rd Qu.:0.03223
Max.   :0.01830   Max.   :0.03526
#CLchatdelt75
summary(delt75[25:36,c(1008,1009)])
     mean               stdev
Min.   :0.01302   Min.   :0.02250
1st Qu.:0.01442   1st Qu.:0.02716
Median :0.01535   Median :0.02948
Mean   :0.01530   Mean   :0.02900
3rd Qu.:0.01609   3rd Qu.:0.03146
Max.   :0.01752   Max.   :0.03473
```

```
#BSuniflambt100
summary(lambt100[37:48,c(1008,1009)])
     mean              stdev
Min.   :0.01081   Min.   :0.01699
1st Qu.:0.01396   1st Qu.:0.02596
Median :0.01514   Median :0.02710
Mean   :0.01495   Mean   :0.02800
3rd Qu.:0.01586   3rd Qu.:0.03162
Max.   :0.01760   Max.   :0.03521
#BSuniflambt200
summary(lambt200[37:48,c(1008,1009)])
     mean              stdev
Min.   :0.01499   Min.   :0.02503
1st Qu.:0.01567   1st Qu.:0.02908
Median :0.01623   Median :0.03220
Mean   :0.01667   Mean   :0.03197
3rd Qu.:0.01792   3rd Qu.:0.03567
Max.   :0.01866   Max.   :0.03677
#BSuniflambt300
summary(lambt300[37:48,c(1008,1009)])
     mean              stdev
Min.   :0.01270   Min.   :0.02164
1st Qu.:0.01431   1st Qu.:0.02652
Median :0.01511   Median :0.02913
Mean   :0.01553   Mean   :0.03061
3rd Qu.:0.01654   3rd Qu.:0.03444
```

```
Max.   :0.01932   Max.   :0.04570
#BSunifrhopsi1
summary(rhopsi1[55:72,c(1008,1009)])
     mean              stdev
Min.   :0.01329   Min.   :0.02417
1st Qu.:0.01544   1st Qu.:0.02906
Median :0.01567   Median :0.03118
Mean   :0.01619   Mean   :0.03217
3rd Qu.:0.01774   3rd Qu.:0.03503
Max.   :0.01932   Max.   :0.04570
#BSunifrhopsi2
summary(rhopsi2[55:72,c(1008,1009)])
     mean              stdev
Min.   :0.01081   Min.   :0.01699
1st Qu.:0.01420   1st Qu.:0.02561
Median :0.01511   Median :0.02731
Mean   :0.01524   Mean   :0.02822
3rd Qu.:0.01646   3rd Qu.:0.03305
Max.   :0.01866   Max.   :0.03591
#BSunifgamet1
summary(gamet1[55:72,c(1008,1009)])
     mean              stdev
Min.   :0.01081   Min.   :0.01699
1st Qu.:0.01505   1st Qu.:0.02620
Median :0.01566   Median :0.02989
Mean   :0.01575   Mean   :0.03019
```

```
3rd Qu.:0.01716   3rd Qu.:0.03455
Max.   :0.01866   Max.   :0.04570
#BSunifgamet2
summary(gamet2[55:72,c(1008,1009)])
     mean             stdev
Min.   :0.01312   Min.   :0.02164
1st Qu.:0.01459   1st Qu.:0.02698
Median :0.01547   Median :0.02927
Mean   :0.01569   Mean   :0.03020
3rd Qu.:0.01696   3rd Qu.:0.03386
Max.   :0.01932   Max.   :0.03801
#BSunifdelt25
summary(delt25[37:48,c(1008,1009)])
     mean             stdev
Min.   :0.01406   Min.   :0.02503
1st Qu.:0.01470   1st Qu.:0.02640
Median :0.01532   Median :0.02817
Mean   :0.01584   Mean   :0.03053
3rd Qu.:0.01663   3rd Qu.:0.03466
Max.   :0.01864   Max.   :0.04570
#BSunifdelt50
summary(delt50[37:48,c(1008,1009)])
     mean             stdev
Min.   :0.01081   Min.   :0.01699
1st Qu.:0.01356   1st Qu.:0.02566
Median :0.01532   Median :0.02820
```

```
Mean   :0.01481   Mean   :0.02733
3rd Qu.:0.01578   3rd Qu.:0.02977
Max.   :0.01794   Max.   :0.03460
#BSunifdelt75
summary(delt75[37:48,c(1008,1009)])
     mean              stdev
Min.   :0.01270   Min.   :0.02219
1st Qu.:0.01555   1st Qu.:0.03132
Median :0.01682   Median :0.03376
Mean   :0.01650   Mean   :0.03272
3rd Qu.:0.01792   3rd Qu.:0.03598
Max.   :0.01932   Max.   :0.03801


#BSchatlambt100
summary(lambt100[49:60,c(1008,1009)])
     mean              stdev
Min.   :0.01360   Min.   :0.02543
1st Qu.:0.01430   1st Qu.:0.02647
Median :0.01587   Median :0.03013
Mean   :0.01600   Mean   :0.03157
3rd Qu.:0.01756   3rd Qu.:0.03650
Max.   :0.01893   Max.   :0.04022
#BSchatlambt200
summary(lambt200[49:60,c(1008,1009)])
     mean              stdev
```

```
Min.   :0.01086   Min.   :0.01756
1st Qu.:0.01482   1st Qu.:0.02928
Median :0.01606   Median :0.03176
Mean   :0.01568   Mean   :0.03033
3rd Qu.:0.01668   3rd Qu.:0.03330
Max.   :0.01869   Max.   :0.03561
#BSchatlambt300
summary(lambt300[49:60,c(1008,1009)])
     mean              stdev
Min.   :0.01225   Min.   :0.02012
1st Qu.:0.01361   1st Qu.:0.02647
Median :0.01476   Median :0.02883
Mean   :0.01494   Mean   :0.02907
3rd Qu.:0.01636   3rd Qu.:0.03197
Max.   :0.01741   Max.   :0.03515
#BSchatrhopsi1
summary(rhopsi1[73:90,c(1008,1009)])
     mean              stdev
Min.   :0.01086   Min.   :0.01756
1st Qu.:0.01392   1st Qu.:0.02741
Median :0.01646   Median :0.03162
Mean   :0.01574   Mean   :0.03076
3rd Qu.:0.01743   3rd Qu.:0.03506
Max.   :0.01893   Max.   :0.04022
#BSchatrhopsi2
summary(rhopsi2[73:90,c(1008,1009)])
```

```
     mean              stdev
Min.   :0.01278   Min.   :0.02476
1st Qu.:0.01389   1st Qu.:0.02629
Median :0.01571   Median :0.02994
Mean   :0.01534   Mean   :0.02989
3rd Qu.:0.01619   3rd Qu.:0.03198
Max.   :0.01869   Max.   :0.03838
#BSchatgamet1
summary(gamet1[73:90,c(1008,1009)])
     mean              stdev
Min.   :0.01086   Min.   :0.01756
1st Qu.:0.01369   1st Qu.:0.02624
Median :0.01547   Median :0.03003
Mean   :0.01512   Mean   :0.02964
3rd Qu.:0.01634   3rd Qu.:0.03261
Max.   :0.01818   Max.   :0.03838
#BSchatgamet2
summary(gamet2[73:90,c(1008,1009)])
     mean              stdev
Min.   :0.01225   Min.   :0.02012
1st Qu.:0.01433   1st Qu.:0.02712
Median :0.01610   Median :0.03114
Mean   :0.01595   Mean   :0.03100
3rd Qu.:0.01731   3rd Qu.:0.03485
Max.   :0.01893   Max.   :0.04022
#BSchatdelt25
```

```
summary(delt25[49:60,c(1008,1009)])
     mean               stdev
Min.   :0.01278   Min.   :0.02476
1st Qu.:0.01375   1st Qu.:0.02823
Median :0.01551   Median :0.02994
Mean   :0.01520   Mean   :0.03044
3rd Qu.:0.01618   3rd Qu.:0.03174
Max.   :0.01893   Max.   :0.04022
#BSchatdelt50
summary(delt50[49:60,c(1008,1009)])
     mean               stdev
Min.   :0.01350   Min.   :0.02527
1st Qu.:0.01499   1st Qu.:0.02645
Median :0.01664   Median :0.03096
Mean   :0.01603   Mean   :0.03116
3rd Qu.:0.01698   3rd Qu.:0.03488
Max.   :0.01869   Max.   :0.03838
#BSchatdelt75
summary(delt75[49:60,c(1008,1009)])
     mean               stdev
Min.   :0.01086   Min.   :0.01756
1st Qu.:0.01382   1st Qu.:0.02604
Median :0.01580   Median :0.03158
Mean   :0.01539   Mean   :0.02937
3rd Qu.:0.01754   3rd Qu.:0.03426
Max.   :0.01860   Max.   :0.03641
```