Project 2

Bieber Fever

By Manu Patil and Katie Barno

In [1]:
```
# Configure Jupyter so figures appear in the notebook
%matplotlib inline

# Configure Jupyter to display the assigned value after an assignment
%config InteractiveShell.ast_node_interactivity='last_expr_or_assign'

# import functions from the modsim.py module
from modsim import *
```

Question: Can the fan and hater populations of a celebrity be controlled entirely by the media?

Assuming the fans of Bieber are, "infected," and using an SIR, or SBR, model to illustrate this phenomenon, how does media influence fans? Boredom? What about those who dislike Bieber simply because of his popularity, the so-called haters? We expand the standard SIR model to account for these two "diseases."

The following cells show our replication of the simulation in the Tweedle-Smith paper. We were able to produce a model that yielded the same results as the paper. From here, we will expand the model to incorporate anti-fans, sweeps of the boredom rate, and a sweep of media coverage from extreme negative to extreme positive.

The Justin Bieber Model was first fully created by Valerie Tweedle and Robert J Smith? (Yes, the question mark is part of his name). The first part of our computational essay attempts to recreate the model proposed by Valerie Tweedle and Robert J. Smith?

What do our variable represent? While, at its core, this model is based on the SIR model, the SBR model has some subtle differences.

S : Susceptible is exactly the same as in the SIR model. It represents the portion of the population who are primed to be infected by the Bieber disease. The Tweedle-Smith? paper describes this stock as a "curious young girl".

B : Bieber- Infected represents the portion of the population who are currently infected with the disease. The Tweedle-Smith? paper describes this stock as a "the same young girl screaming at a concert".

R : Recovered is slightly different in comparison to the SIR model. This model considers the fact that "immunity" is not a an end state. A member of the model may be infected again and proceed through the entire cycle again. These flows are described in more thorough detail below. The Tweedle-Smith? paper describes this stock as a "a sullen teenager who won't listen to a word you say, but still wants you to take them to the mall".

Let us als explore the parameters that govern our system:

ϵ : The model assumes that total Media Consumption is represented by M. Epsilon is the ratio of positive media to negative media. It is unitless.

π : The 'recruitment rate' here refers to the rate at which young people enter the susceptible category from outside of the system. It is measured in people month^{-1}

μ : The 'maturation rate' represents the rate at which people age out of our system. We are modeling individuals between the ages of five and seventeen years old. With a model lifespan of twelve years, we assume that a 12 of the population matures out of the system each year.

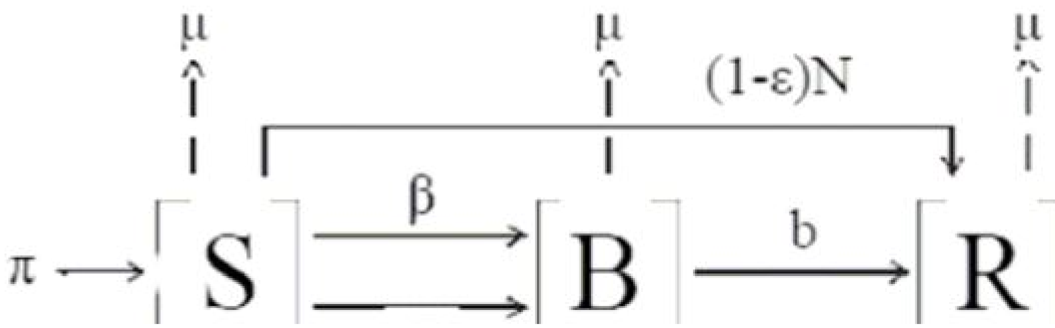
β : The 'transmission rate' is the rate at which susceptible individuals become fans. It is measured in people $^{-1}$month^{-1} as people become infected as they come in contact with those who are infected.

b : Our 'boredom rate' is the rate of recovery from the bieber disease. This rate is not necessarily due to negative media, fans have just lost interest. It is measured in month^{-1} with larger values indicated faster boredom.

P : The 'positive media rate' is the number of positive media events surrounding Bieber in a month. This can include album releases, singles, books, or movies. This is measured in month^{-1}

N : The 'negative media rate' is the number of negative media events surrounding Bieber in a month. This considers gossip magazines attempting to slander Bieber's image.

The media flows make the model slightly more complicated. To be explicit, positive media can cause a recovered individual to move to the susceptible category or moved a susceptible individual to the Bieber-Infected Category. Negative Media can cause a Bieber-Infected individual to move to the susceptible category or cause a susceptible individual to move to the Recovered category. The Figure below illustrates the model:



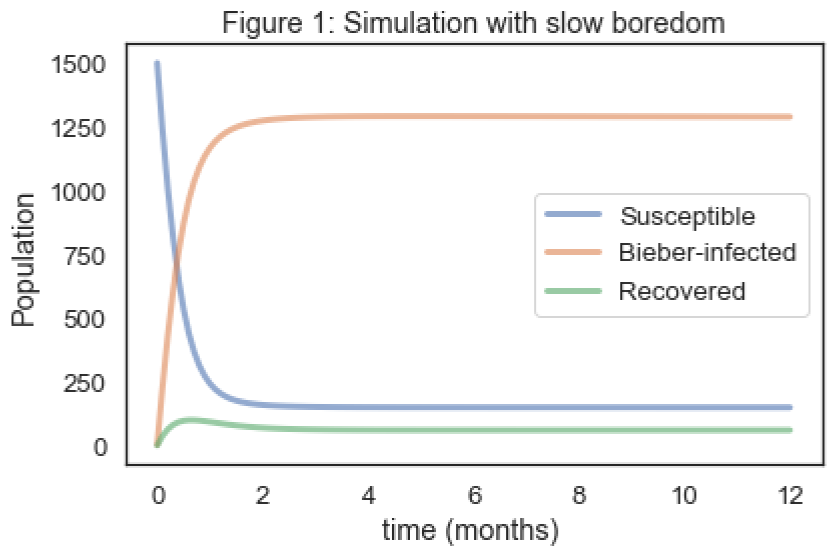
Model #1

In [2]:
```
def make_system(pi, Beta, mu, b ,P, N, epsilon, dt):
    """Make a system object for the SBR model.
    S: Susceptible
    B: Bieber-infected
    R: Recovered
    pi: Recruitment Rate
    Beta: Transmission Rate
    mu: Maturation Rate
    b: Boredom Rate
    P: Positive Media Rate
    N: Negative Media Rate
    epsilon: Positive Media Proportion

    returns: System object
    """
    init = State(S = 1500, B = 3, R = 0);

    t0 = 0;
    t_end = 12; #chosen to simulate 1 year

    return System(init=init, t0=t0, t_end=t_end,
                  pi =pi,
                  Beta = Beta,
                  mu = mu,
                  b  = b,
                  P = P,
                  N = N,
                  epsilon= epsilon,dt=dt);
```

In [3]:
```
def update_func(state, t, system):
    """Update the SBR model.

    state: State (S,B,R)
    t: time
    system: System object

    returns: State (S,B,R)
    """
    unpack(system)
    S,B,R = state;
    entranceRate = pi # people that enter the system per month
    infectedRate = Beta*S*B # number of people infected each month
    posMediaPercent = epsilon*P
    negMediaPercent = (1-epsilon)*N
    boredom = b*B
    ds = entranceRate - infectedRate - posMediaPercent*S + negMediaPercent*B +
 posMediaPercent*R - negMediaPercent*S - mu*S
    db = infectedRate + posMediaPercent*S - negMediaPercent*B - boredom - mu*B

    dr = boredom - posMediaPercent*R + negMediaPercent*S - mu*R
    S += ds* dt
    B += db*dt
    R += dr*dt

    return State(S = S, B = B, R = R);
```

In [4]:
```
def run_simulation(system, update_func):
    """Runs a simulation of the system.

    system: System object
    update_func: function that updates state

    returns: TimeFrame
    """
    unpack(system)
    frame = TimeFrame(columns=init.index)
    frame.row[0] = init

    for t in linrange(t0, t_end, dt):
        frame.row[t+dt] = update_func(frame.row[t], t, system)

    return frame
```

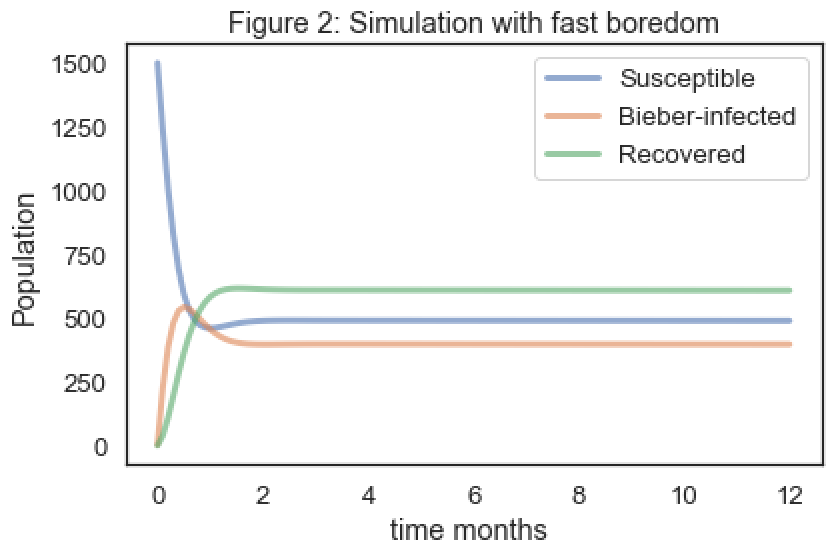
In [5]:
```
system = make_system(10, .00083, 1/144,1/24 , 2, 1, .75,.1)
frame = run_simulation(system, update_func);
plot(frame.S, label  = "Susceptible");
plot(frame.B, label = "Bieber-infected");
plot(frame.R, label = "Recovered");

decorate(xlabel = "time (months) ", ylabel = "Population", title = "Figure 1:
 Simulation with slow boredom");
```



Results 1a

The above graph displays the model we replicated from the original paper. We produced the same results as the paper for this boredom rate, 1/24. The results are similar to those published in the paper.

In [6]:
```
system = make_system(10, .00083, 1/144,2 , 2, 1, .75,.1)#Given Parameters and
 dt set to 1/24 such that time is represented in hours
frame = run_simulation(system, update_func);
plot(frame.S, label  = "Susceptible");
plot(frame.B, label = "Bieber-infected");
plot(frame.R, label = "Recovered");

decorate(xlabel = "time months ", ylabel = "Population", title = "Figure 2: Si
mulation with fast boredom");
```



Results 1b

This graph shows the same model as the one in the graph above, only here we have increases the boredom from 1/24 to 2. Again, we plotted this graph to show that we have successfully replicated the simulations we saw in the paper. In this version of the model, the high rate of boredom

For the sake of comparison later, we want note that in the model with slower boredom the diseased population approaches a higher equilibrium than those of the susceptible or recovered populations. We notice the opposite effect with the fast boredom case.

Model #2: Introduction of Haters

Due to his enourmous popularity, Justin Bieber has gathered a number of people who dislike him for no reason other than to rebel against what they percieve to be the mainstream. These "haters" act almost as "infected" as our Bieber fans. The difference is, the hater population is dependant on the fan population. We expand our model from three stocks, S,B, and R; to four, S, B, R, and H.

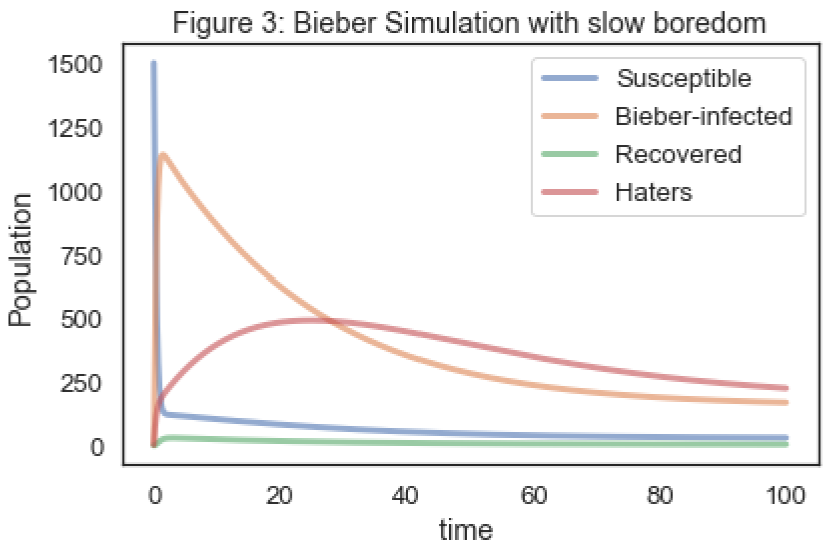
In [7]:
```
def make_system_H(pi, Beta, mu, b ,P, N, epsilon, gamma, dt):
    """Make a system object for the SBR model.
    S: Susceptible
    B: Bieber-infected
    R: Recovered
    pi: Recruitment Rate
    Beta: Transmission Rate
    mu: Maturation Rate
    b: Boredom Rate
    P: Positive Media Rate
    N: Negative Media Rate
    epsilon: Positive Media Proportion
    gamma = Hater rate

    returns: System object
    """
    init = State(S = 1500, B = 3, R = 0, H = 3);

    t0 = 0;
    t_end = 100;

    return System(init=init, t0=t0, t_end=t_end,
                  pi =pi,
                  Beta = Beta,
                  mu = mu,
                  b  = b,
                  P = P,
                  N = N,
                  epsilon= epsilon,
                  gamma = gamma, dt=dt);
```

The function "make_system_H" makes a similar system to the original one. The difference is the parameter gamma, which controls the rate at which the suceptible population become "haters."

In [8]:
```
def update_func_H(state, t, system):
    """Update the SBR model.

    state: State (S,B,R)
    t: time
    system: System object

    returns: State (S,B,R)
    """
    unpack(system)
    S,B,R,H = state;
    entranceRate = pi # people that enter the system per month
    infectedRate = Beta*S*B # number of people infected each month
    posMediaPercent = epsilon*P
    negMediaPercent = (1-epsilon)*N

    ds = entranceRate-infectedRate-posMediaPercent*S+negMediaPercent*B + posMe
diaPercent*R - negMediaPercent*S - mu*S -gamma*B
    db = infectedRate + posMediaPercent*S - negMediaPercent*B - b*B - mu*B
    dr = b*B - posMediaPercent*R - mu*R
    dh = negMediaPercent*S + gamma*B - b*H - mu*H

    S += ds*dt
    B += db*dt
    R += dr*dt
    H += dh*dt

    return State(S = S, B = B, R = R, H = H);
```

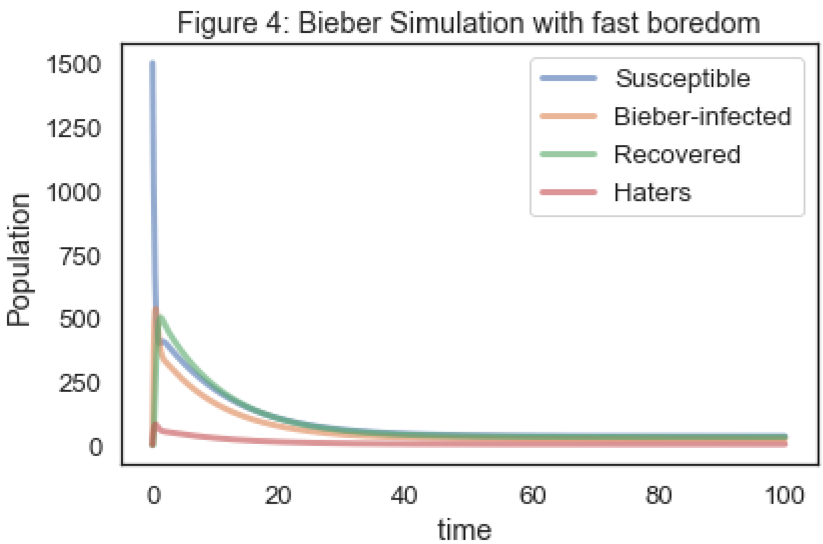
This new update function included a new differential equation for our hater population. Haters come from suceptible due to negative media coverage relating to Bieber, this rate is represented by the "negMediaPercent*S" term, or they become haters because they simply object to Biebers following, "gamma*B". People leave the hater population when they become bored of their haterdom, "b*H", or when they age out of our system, "mu*H".

In [9]:
```
system = make_system_H(10, .00083, 1/144, 1/24 , 2, 1, .75, 0.01, .1)
frame = run_simulation(system, update_func_H);

plot(frame.S, label  = "Susceptible");
plot(frame.B, label = "Bieber-infected");
plot(frame.R, label = "Recovered");
plot(frame.H, label = "Haters");
decorate(xlabel = "time ", ylabel = "Population", title = "Figure 3: Bieber Si
mulation with slow boredom");
```



The model above shows the incorporation of Bieber Haters.

In [10]:
```
system = make_system_H(10, .00083, 1/144, 2, 2, 1, .75,.01,.1)
frame = run_simulation(system, update_func_H)
plot(frame.S, label  = "Susceptible");
plot(frame.B, label = "Bieber-infected");
plot(frame.R, label = "Recovered");
plot(frame.H, label = "Haters");

decorate(xlabel = "time ", ylabel = "Population", title = "Figure 4: Bieber Si
mulation with fast boredom" );
```



Results #2

To verify this model, we compare the results to the model without Bieber Haters. The same basic trend is found to be followed. With slower boredom the diseased populations approach a higher equilibrium than those of the susceptible or recovered populations. We notice the opposite effect with the fast boredom case.

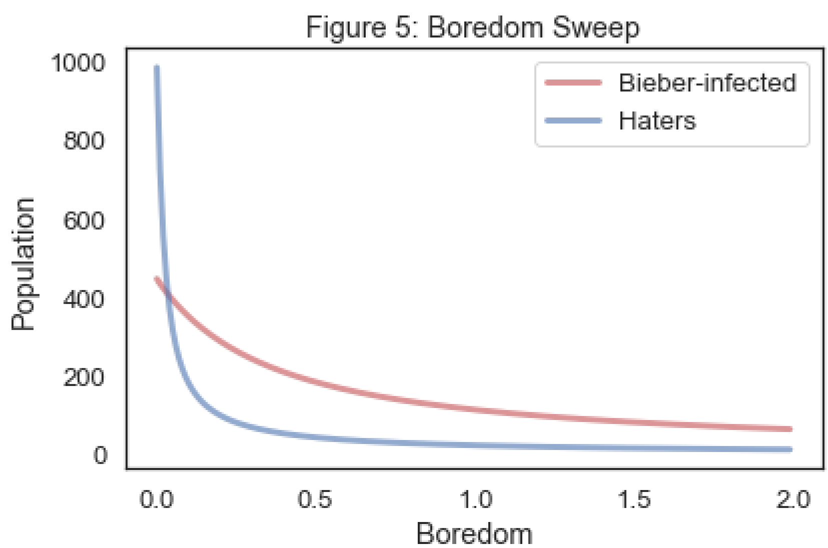
Sweeping Boredom

Below we begin our first parameter sweep. Boredom controls the retention of the fan and hater populations. As boredom is increased, we expect to see both fans and haters leave their respective groups more quickly.

In [11]:
```
def sweep_b(pi, Beta, mu, b_array ,P, N, epsilon, gamma, dt):
    """Sweep a range of values for b.

    beta_array: array of boredom values

    returns:
    sweepB: sweepSeries that maps from b to max infected with main disease
    sweepH: sweepSeries that maps from b to max infected with hater disease
    """
    sweepB = SweepSeries()
    sweepH = SweepSeries()

    for b in b_array:
        system = make_system_H(pi, Beta, mu, b ,P, N, epsilon, gamma, dt)
        frame = run_simulation(system, update_func_H)
        sweepB[system.b] = frame.B.mean()
        sweepH[system.b] = frame.H.mean()


    return sweepB, sweepH;
```

In [12]:
```
boredom_array = linrange(0,2,.01)
sweepB, sweepH = sweep_b(10, .00083, 1/144, boredom_array, 2, 1, .75,.01,.1);
```

In [13]:
```
plot(sweepB, label = "Bieber-infected",color = "r");
plot(sweepH, label = "Haters",color = "b");
decorate(xlabel = "Boredom", ylabel = "Population", title = "Figure 5: Boredom
 Sweep");
```



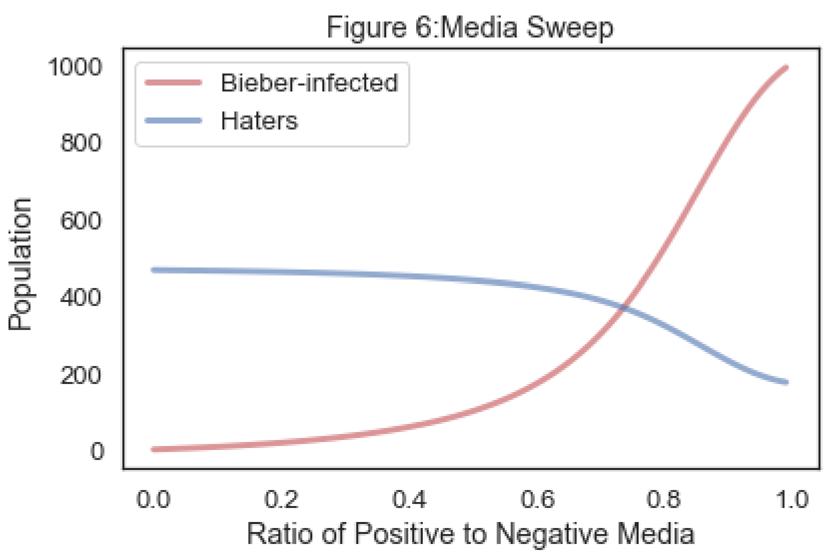
This graph outputs the expected results of having the average population drop down as boredom increases.

Sweeping the Ratio of Positive/Negative Media

In [14]:
```
def sweep_Media_Ratio(pi, Beta, mu, b ,P, N, e_array, gamma, dt):
    """Sweep a range of values for e.

    e_array: array of epsilon values


    returns:
    sweepB: sweepSeries that maps from b to max infected with main disease
    sweepH: sweepSeries that maps from b to max infected with hater disease
    """
    sweepB = SweepSeries()
    sweepH = SweepSeries()
    for e in e_array:
        system = make_system_H(pi, Beta, mu, b ,P, N, e,gamma, dt)
        frame = run_simulation(system, update_func_H)
        sweepB[system.epsilon] = frame.B.mean()
        sweepH[system.epsilon] = frame.H.mean()

    return sweepB, sweepH;
```

In [15]:
```
e_array = linrange(0,1,.01);
sweepB,sweepH = sweep_Media_Ratio(10, .00083, 1/144, 1/24, 2, 1, e_array,.01,.
1);
```

In [16]:
```
plot(sweepB, label = "Bieber-infected",color = "r");
plot(sweepH, label = "Haters",color = "b");

decorate(xlabel = "Ratio of Positive to Negative Media", ylabel = "Population"
, title = "Figure 6:Media Sweep");
```

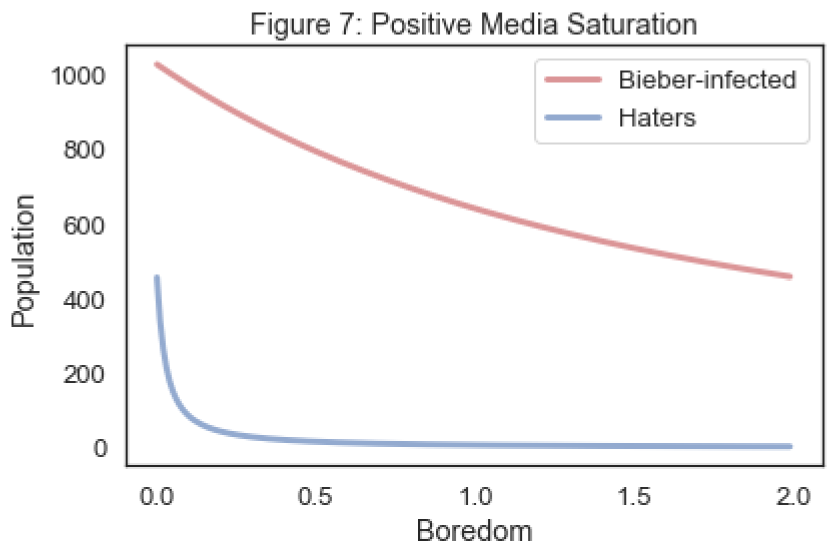


This graph, showing the relationship between media and hater population, follows generally what we would expect. When the ratio of positive to negative media is under .5 the hater population thrives on the negative media. After crossing .5 the Bieber infected population surges with the increase in positive media. The Haters continue around the same level thriving off of the elavated Bieber fan levels. However, with the lack of much negative news their population begin to drop off towards complete Media saturation.

Boredom Sweep with Postive Media saturation. In this sweep we will try to maximize the Bieber-Infected population while minimizing the hater population. We expect that the boredom will only have a multiplying effect

In [17]:
```
boredom_array = linrange(0,2,.01)
sweepB1, sweepH1 = sweep_b(10, .00083, 1/144, boredom_array, 2, 1, 1,.01,.1);
```

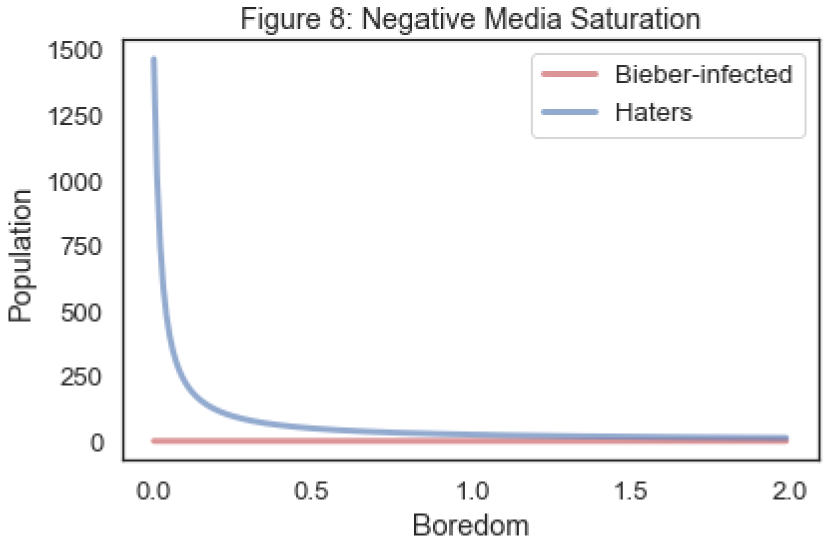
In [18]:
```
plot(sweepB1, label = "Bieber-infected",color = "r");
plot(sweepH1, label = "Haters",color = "b");
decorate(xlabel = "Boredom", ylabel = "Population",title = "Figure 7: Positive
 Media Saturation");
```



Above is a Boredom Sweep with Negative Media Saturation. In this sweep, we expect that nearly both populations will die off as boredom increases. First, the Bieber-fans will simply up and leave. Continous bad news disrupts fan loyalty. For the haters, at slow levels of boredom, they will thrive and their population will be maximized because of the negative media feeding their population, however, at faster levels they will die out as well.

In [19]:
```
boredom_array = linrange(0,2,.01)
sweepB2, sweepH2 = sweep_b(10, .00083, 1/144, boredom_array, 2, 1, 0,.01,.1);
```

In [20]:
```
plot(sweepB2, label = "Bieber-infected",color = "r");
plot(sweepH2, label = "Haters",color = "b");
decorate(xlabel = "Boredom", ylabel = "Population", title = "Figure 8: Negativ
e Media Saturation");
```



The results of these two sweeps suggest some interesting conclusions about control over a fans populations. With constant postive media saturation, we can maximize the Bieber fans, while mitigating the negative effect of the haters. With constant negative media saturation, we can maximize Bieber haters.

Why is this important? Because we can make money off of these people. Consider the revenue flows of fans purchasing merchandise or haters purchasing merchandise that makes it clear of their hate for Bieber

Interpretation

We set out looking for the effects of media on the fan and hater populations of Justin Bieber. Clearly the media influences public opinion, but to what extent? Could a world with no media still contain a high number of Bieber fans? Could complete positive media decimate the hater population? To what degree does boredom influence the fan retention rate?

We found some interesting results. A sweep of our positive to negative media ratio demonstrates how at higher positive media levels the fans grow in numbers while the hater population shrinks. Similarly, a disproportionate rate of negative media increases the hater population while it shrinks the fanbase.

We were more surprised by the results of our boredom sweep, which convinced us that the media is actually the deciding factor in controlling fan populations. In a world saturated with positive media, our boredom sweep showed that increasing boredom decreased the populations of both fans and haters, but the hater population was virtually at most values of boredom. Only if boredom is incredibly low can Bieber haters exist in a positive-Bieber-media world. The boredom sweep in a world completely filled with negative Bieber media showed similar results. In this world, Bieber does not have fans. Because the hater population is influenced by the fan population, if the boredom gets larger than a small fraction there will not be any haters. We were surprised to find that the hater population might be zero in a world that only had negative media. However, this situation might be more realistic because our model assumes boredom works quickly when in reality it might be a much slower force.

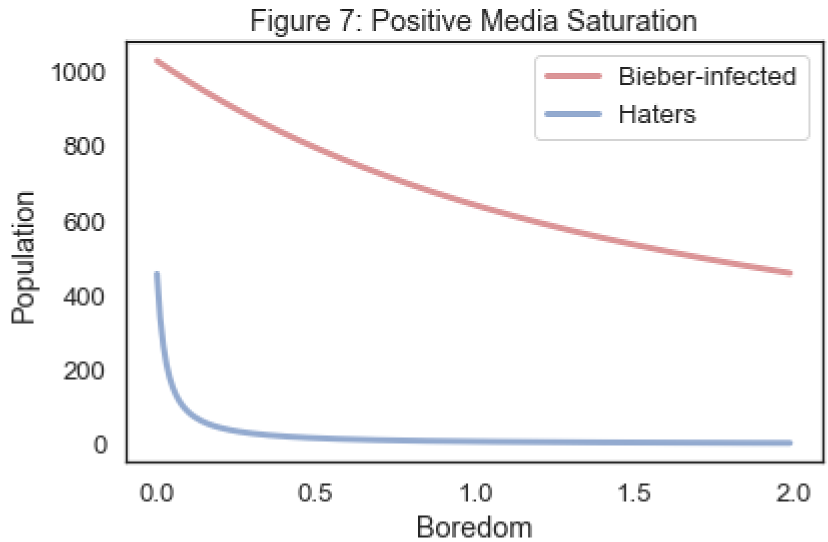
Boredom doesn’t effect the fans and haters in different ways. When media is positive at any boredom, there are fewer Bieber haters than Bieber fans. At and boredom with only negative media there will be more Bieber haters than fans. Boredom does not determine which population is greater than the other, it can only scale both population sizes.

After considering our results we think they make sense. The media had the effect we thought it would. We did think that boredom would have more of an effect than it did. If we assume that boredom effects Bieber fans at the same rate as Bieber haters, and we do, this model works.

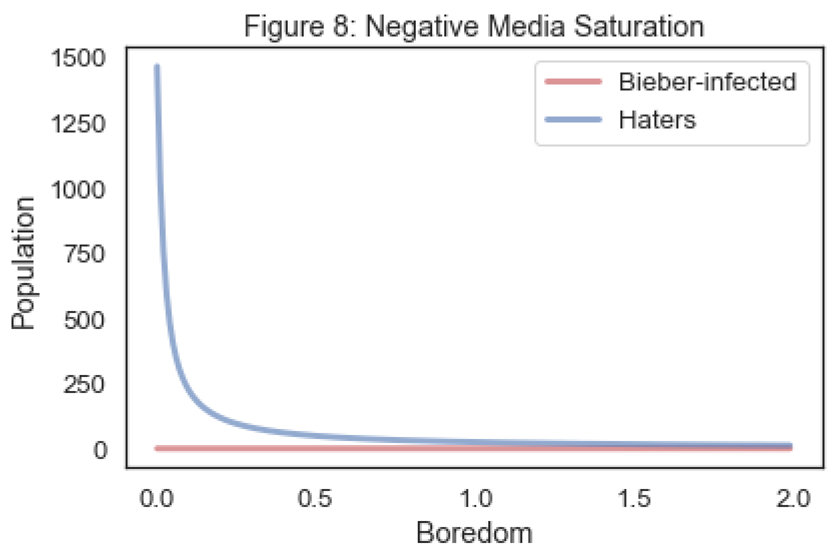
Abstract

We asked if the fan and hater populations of a celebrity be controlled entirely by the media. Our short answer is yes. The fans are directly controllable by the media. However, because the haters are influenced by both the media and the fan population they are harder to directly control. This is shown by Figure 7 and Figure 8 displayed below. Why is data important? As mentioned earlier, control over these groups can create extreme profits. By manipulating the quantity and loyalty of fans, one can effectively control both sides of the market.

In [21]:
```
plot(sweepB1, label = "Bieber-infected",color = "r");
plot(sweepH1, label = "Haters",color = "b");
decorate(xlabel = "Boredom", ylabel = "Population", title = "Figure 7: Positiv
e Media Saturation");
```



In [22]:
```
plot(sweepB2, label = "Bieber-infected",color = "r");
plot(sweepH2, label = "Haters",color = "b");
decorate(xlabel = "Boredom", ylabel = "Population", title = "Figure 8: Negativ
e Media Saturation");
```



Works Cited

Tweedle, Valerie, and Robert J. Smith? “A Mathematical Model of Bieber Fever: The Most Infectious Disease of Our Time?”