Lab #6 Disease control
Bio 347 - Disease Ecology

**Objectives:**

- Determine the necessary vaccination or culling effort needed to eliminate a parasite transmitted with density dependence
- Learn how to use for loops to easily accomplish this in R

**Background:**

You are a state manager of lakes in the upper Midwest that host some of the best trout fishing in the US. Recently, a virulent parasite has arrived in your lakes, and it is threatening your tourism industry. Anglers dislike catching trout infected with cestode worms because they see them while they clean their fish, get grossed out, and then find a "better" place to spend their fishing trips (FYI, Google searching "worms in trout" leads you to message boards where people complain about this and worry that they themselves could become infected). You found out that these cestodes seem to be transmitted by typical direct contacts, and so you are considering culling as a management option. Fortunately, scientists at the FWS have developed an injectable vaccine that can provide relatively long-lasting immunity. Both strategies might be effective, so now you want to evaluate whether they can boost the density of uninfected trout in your lakes (which you typically estimate from 1000 L (1 kL) net hauls), which should entice anglers to return.

Susceptible-Infected (SI) models can be modified to include representations of various management strategies. We saw in class that vaccination and culling can reduce parasite spread by reducing the density of susceptible hosts (both vaccination and culling) and the infectious period (just culling). Here we will examine the ability of these two approaches to control a parasite that is transmitted with density dependence. In order to accomplish this more easily, we will use *for* loops in R. Loops are easy to write and interpret. They are slow, but this

drawback is only relevant for large problems that take more than a few hours to run, so it will not be a problem for us.

## *for* loops

There are many cases where we want R to do the same thing (or very nearly the same thing) more than once. We can write out each *iteration*, however, this becomes very tedious if we want to *iterate* our code many times. In that case, we can write a *for loop* to complete a task a specified number of times. Here is an example:

```
# This loop simply displays ("prints") the iteration count each time
for ( i in 1:100){ # inside the loop, we can use i, which loops over values from 1 to 100
    print(i)
}
```

You can also use loops to store a result from a calculation you want to repeat many times:

```
# This loop stores the squares of each number
squares = numeric() # First, we define an empty vector
for ( i in 1:100){
    squares[i] = i^2
}
squares # Check that it worked
```

This above example works well because we can index into a new position in the variable squares with the syntax, squares[i]. If you are not yet comfortable with the concept of *indexing*, please Google the "R short reference card".

Here is a basic refresher on indexing: Use a number inside square brackets, [*number*], immediately after the name of a variable to indicate specific data stored in that position. For example, squares[4] would return the fourth value stored in the vector squares. You can index into lists (which we really don't use) and vectors, which we often use, this way. You can also index into data frames and matrices, two-dimensional data structures, with a two number [*row*,*column*] system: output[1,2] would return the value stored in the first row and second column of the matrix or data frame named output.

You can get entire rows (e.g., output[1,]) or entire columns (e.g., output[,1]) by leaving the other field empty.

**Questions**

1) (**1 point**) Specify the following model as a function called vaccination_and_culling, so that you can simulate it with lsoda:

$$\frac{dS}{dt} = b_M\big(1 - c(S+I+R)\big) * (S+I+R) - (d+k+\gamma)S - \beta SI + \omega R$$
$$\frac{dI}{dt} = \beta SI - (d+v+k)I$$
$$\frac{dR}{dt} = \gamma S - (\omega + d + k)R$$

This model tracks the changes in densities for three types of hosts, susceptible, *S*, infectious, *I*, and resistant, *R*. Susceptible hosts increase from density-dependent births, with maximum rate $b_M$, and strength of competition, *c*. Susceptible hosts are lost to background death, *d*, and culling, *k*, density-dependent transmission, at rate β, and vaccination, at rate γ. Susceptible hosts also increase through the waning of immunity of resistant hosts, at rate ω. Infectious hosts increase due to transmission, and they decrease due to background death, *d*, parasite virulence, *v*, and culling, *k*. Resistant individuals increase due to vaccination, at rate γ, and they decrease due to the waning of immunity.

2) (**1.5 points**) First, evaluate the effect of vaccination in the absence of any culling. Use the following parameters: ($b_M$ = 0.5 month$^{-1}$, $d$ = 0.05 month$^{-1}$, $c$ = 0.01 kL, β = 0.01 kL month$^{-1}$, ω = 0.05 month$^{-1}$, $v$ = 0.15 month$^{-1}$, $k$ = 0 month$^{-1}$). Simulate this model with initial conditions (S=99 kL$^{-1}$, I=1 kL$^{-1}$, R=0 kL$^{-1}$) over time = 0:100 for 101 different values of γ, ranging from 0 to 1 in increments of 0.01. At the end of each simulation, store the value S + R, which reflects the total density of all individuals that are alive and not infected at equilibrium, and the value of I. Plot these densities against the value of γ and explain in a comment the level of vaccination effort, γ, required to eliminate the parasite. Hint: use the seq() function to generate a vector for these values of γ.

3) (**1.5 points**) Now, evaluate the effect of culling in the absence of any vaccination. Use the following parameters: ($b_M$ = 0.5 month$^{-1}$, $d$ = 0.05 month$^{-1}$, $c$ = 0.01 kL, β = 0.01 kL month$^{-1}$, ω = 0.05 month$^{-1}$, $v$ = 0.15 month$^{-1}$, γ = 0 month$^{-1}$). Simulate this model with initial conditions (S=99, I=1, R=0) over time = 0:100 for 101 different values of $k$, ranging from 0 to 1 in increments of 0.01. At the end of each simulation, store the value of S, which reflects the total density of all individuals that are alive and not infected at equilibrium, and the value of I. Plot these densities against the value of $k$ and explain in a comment the level of culling effort, $k$, required to eliminate the parasite. Explain which strategy you would use if cost is not a concern.

4) (**1 point**) Use a loop to examine the effects of both culling and vaccination. Use the same parameters as above for $b_M$, $d$, $c$, $v$, β, and ω, but now vary vaccination and culling and record the density of uninfected hosts, S + R, at the end of each simulation. Hint: this can all be done in a single loop by repeating the vectors that you've created for γ and $k$:

New_gamma_vector = rep(gamma_vector, times=101)
New_k_vector = rep(k_vector, each=101)

These new vectors are organized in such a way that New_gamma_vector[i], New_k_vector[i] loops over all possible combinations of γ and $k$ over 101x101 = 10,201 simulations. Plot the results in a contour plot using the following code:

```
# You will need to convert your long vector "S_plus_R" to a matrix
output = matrix(S_plus_R, nrow=101, ncol=101, byrow=T)
# This just helps with the correct shading
mylevels = seq(0,100, 10)
# This is tricky: x and y must be the unrepeated vectors, and z must be the matrix
# If you've gotten this far and still have problems, ask us ASAP.
filled.contour(x=k_vector, y=gamma_vector, z=output, levels = mylevels)
```

Explain in a comment what type of strategy you would use (purely culling, purely vaccination, a mix of the two, neither, or something else) and why?