

Assignment 2

Clint Valentine

Effective Biocontrol Agents

We will simulate the dynamics of the Rosenzweig-MacArthur predator-prey dynamics to illustrate the effectiveness of each predator type as a biocontrol for an invasive pest species. We first simulate the species for continuous logistic growth to confirm the equilibrium abundance of the invasive pest species. We then set up a system of coupled differential equations for each scenario and solve them using the `ode` solver in R.

To measure the impact each predator has on species abundance we average the abundances of the prey during the last 500 timesteps of a 3,000 timestep simulation. We then take the log average of these values for varying predator handling time and plot the results.

No Predator

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right)$$

Type I

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right) - aNP$$

$$\frac{dP}{dt} = abNP - mP$$

Type II

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right) - \frac{aNP}{1 + ahN}$$

$$\frac{dP}{dt} = \frac{abNP}{1 + ahN} - mP$$

Type III

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right) - \frac{aN^2P}{1 + ahN^2}$$

$$\frac{dP}{dt} = \frac{abN^2P}{1 + ahN^2} - mP$$

```

source('http://faraway.neu.edu/data/cb.R')
library('deSolve')

time = 1:3000
handling = seq(0, 1, length=20)

solve.logistic = function(t, y, parms) {
  return(list(parms$rPrey * y[1] * (1 - (y[1] / parms$KPrey))))
}

solve.pred.preyI = function(t, y, parms) {
  dN <- parms$rPrey * y[1] * (1 - (y[1] / parms$KPrey)) - (parms$a * y[1] * y[2])
  dP <- (parms$a * parms$b * y[1] * y[2]) - parms$m * y[2]
  return(list(c(dN, dP)))
}

solve.pred.preyII = function(t, y, parms) {
  dN <- parms$rPrey * y[1] * (1 - (y[1] / parms$KPrey)) -
    (parms$a * y[1] * y[2]) / (1 + parms$a * parms$h * y[1])
  dP <- (parms$a * parms$b * y[1] * y[2]) /
    (1 + parms$a * parms$h * y[1]) - parms$m * y[2]
  return(list(c(dN, dP)))
}

solve.pred.preyIII = function(t, y, parms) {
  dN <- parms$rPrey * y[1] * (1 - (y[1] / parms$KPrey)) -
    (parms$a * (y[1] ** 2) * y[2]) / (1 + parms$a * parms$h * (y[1] ** 2))
  dP <- (parms$a * parms$b * (y[1] ** 2) * y[2]) /
    (1 + parms$a * parms$h * y[1]) - parms$m * y[2]
  return(list(c(dN, dP)))
}

compute.log.response = function(pre, KPrey) {
  return(log(mean(pre/KPrey)))
}

response.ratio = matrix(nrow=length(handling), ncol=3,
  dimnames=list(NULL, c('I', 'II', 'III')))

for (i in 1:length(handling)) {
  parms = list(rPrey=3, KPrey=10, a=2, b=1, m=1.5, h=handling[i])

  #N.equil = parms$m / (parms$a * (parms$b - parms$h * parms$m))

  #P.equil = (parms$rPrey / (parms$a * parms$KPrey)) *
  #      (parms$KPrey + N.equil * (parms$a * parms$h * parms$KPrey - 1) -
  #      parms$a * parms$h * (N.equil ** 2))

  N = c(0.1, 0.1)

  without.predator = ode(y=N[1], times=time, func=solve.logistic, parms, method='ode45')
  with.predatorI = ode(y=N, times=time, func=solve.pred.preyI, parms, method='ode45')
  with.predatorII = ode(y=N, times=time, func=solve.pred.preyII, parms, method='ode45')
}

```

```

with.predatorIII = ode(y=N, times=time, func=solve.pred.preyIII, parms, method='ode45')

response.ratio[i, 'I'] = compute.log.response(tail(with.predatorI[,2], 500),
                                              tail(without.predator[,2], 500))

response.ratio[i, 'II'] = compute.log.response(tail(with.predatorII[,2], 500),
                                              tail(without.predator[,2], 500))

response.ratio[i, 'III'] = compute.log.response(tail(with.predatorIII[,2], 500),
                                              tail(without.predator[,2], 500))
}

```

The plot below shows an expected result. A type I predator, as described by the Lotka-Volterra model, is not described with a function that considers handling time. Instead, this predator should have a constant relationship with its affect on prey abundance.

A type II predator incorporates handling time and begins, at low handling time, like the type I predator. As handling time increases the predator becomes very inefficient at affecting the prey negatively. As handling time increases to a higher amount the predator loses all efficiency and has very little to no negative effect on the prey.

The type III predator is slightly less effective than the type I predator at near zero handling time. The type III predator maintains a negative impact on the prey throughout all ranges of handling time but performs worse than the type I predator as handling time increases.

The type I predator appears to have the most stable and impactful negative effect on the prey over the entire range of handling times. The second most able predator is the type III predator which performs slightly worse than that of the type I predator due to the consideration of handling time in the model of the type III predator. Finally, the type II predator performs the worst by far and loses all effectiveness at high handling times.

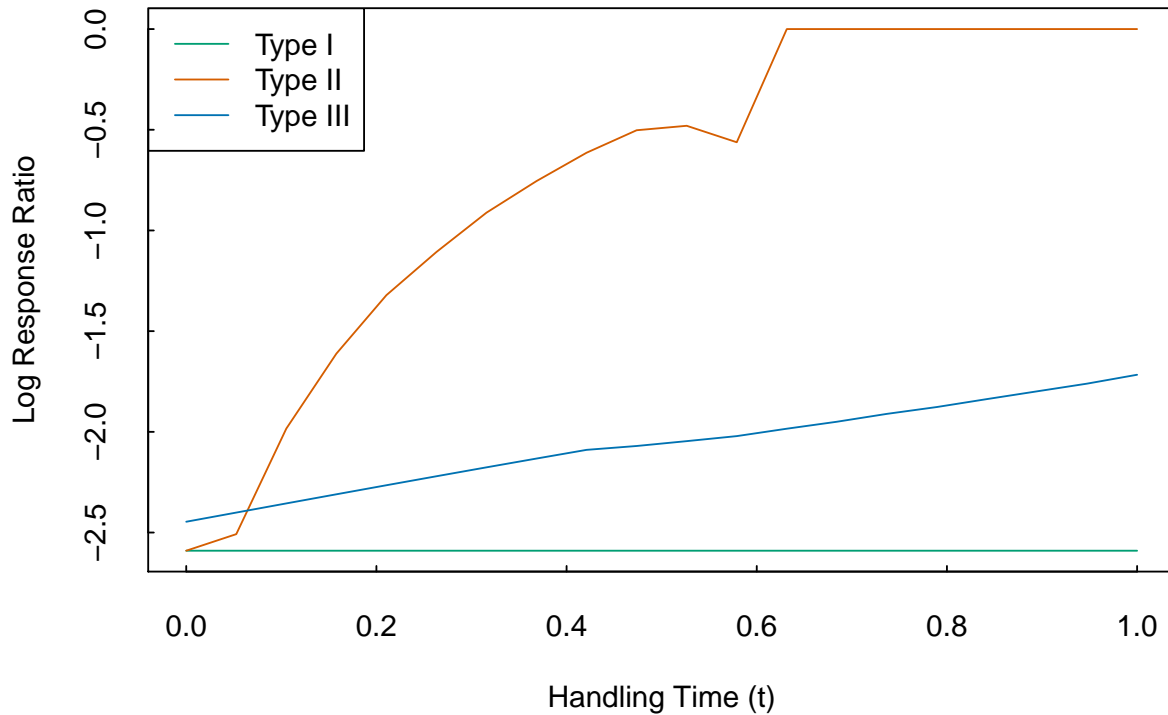
```

matplot(handling, response.ratio[,1:3], type='l', lty=1, col=c(cb.green, cb.red, cb.blue),
        labels=TRUE, tck=0.01, xlab='Handling Time (t)', ylab='Log Response Ratio',
        main='Log Response Ratio vs. Handling Time for Predator Type')

legend('topleft', legend=c('Type I', 'Type II', 'Type III'),
      col=c(cb.green, cb.red, cb.blue), lty=rep(1, 3))

```

Log Response Ratio vs. Handling Time for Predator Type



To determine if the addition of any of the three predator types may destabilize the system we will run an enrichment experiment for predatory-prey systems with carrying capacities K ranging from 2 to 14 in 20 linearly-spaced values. For each value of K we will measure the local min/max of the prey abundance over the last 500 time steps of the simulation and plot the results. We will hold the handling time for the predator constant at $h = 0.5$.

```
time = 1:3000
carry = seq(2, 14, length=40)

solve.forK = function(K, FUN) {
  N = c(0.1, 0.1)
  parms = list(rPrey=3, a=2, b=1, m=1.5, h=0.5, KPrey=K)

  a = ode(y=N, times=time, func=FUN, parms, method='ode45')

  values = numeric()

  for (x in (length(a) - 500):(length(a) - 1)) {
    locmin = (a[x] < a[x - 1]) & (a[x] < a[x + 1])
    locmax = (a[x] > a[x - 1]) & (a[x] > a[x + 1])
    stability = (a[x] == a[x - 1]) & (a[x] == a[x + 1])

    # Append to found values
    if (locmin | locmax | stability) {
      values <- c(values, a[x])
    }
  }
}
```

```

    }
  }
  return(unique(values))
}

plot(NULL, xlim=c(min(carry), max(carry)), ylim=c(0, 9), tck=0.01,
      ylab='Local Min. and Max. of Prey Abundance (N)', xlab='Carrying Capacity (K)',
      main='Stability Analysis of Prey vs. Prey Carrying Capacity')

legend(x='topleft', legend=c('Type I', 'Type II', 'Type III'),
       pch=16, col=c(cb.green, cb.red, cb.blue))

abline(h=0, lty=2, col=cb.grey)

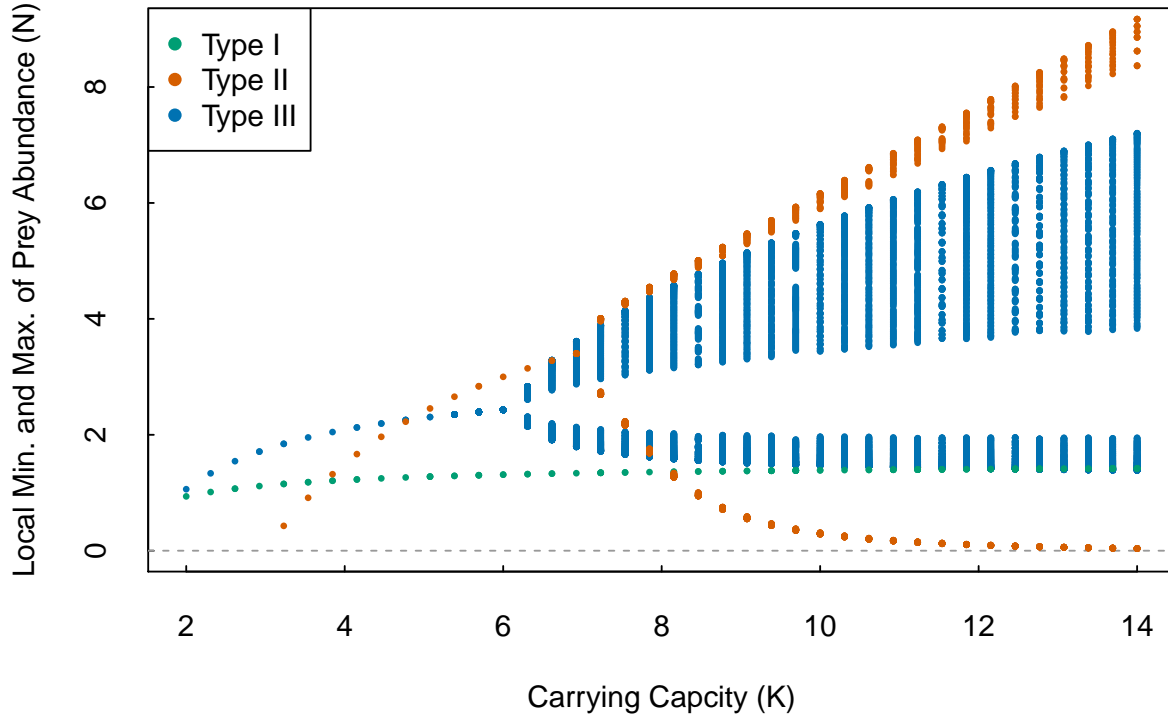
for (K in carry) {
  # Plot Type III first since many overlapping points
  Y = solve.forK(K, solve.pred.preyIII)
  X = rep(K, length.out=length(Y))
  points(X, Y, pch=16, cex=0.5, col=cb.blue)

  Y = solve.forK(K, solve.pred.preyI)
  X = rep(K, length.out=length(Y))
  points(X, Y, pch=16, cex=0.5, col=cb.green)

  Y = solve.forK(K, solve.pred.preyII)
  X = rep(K, length.out=length(Y))
  points(X, Y, pch=16, cex=0.5, col=cb.red)
}

```

Stability Analysis of Prey vs. Prey Carrying Capacity



This analysis shows that only a type I predator is able to maintain a stable predator-prey system as shown by the linear shape of the points on the above plot. This predator is thus least useful as a biocontrol in a predator-prey relationships as the prey will always be able to maintain a stable existence. We also see that both the type II and type III predators destabilize the abundance of prey. This destabilizing effect increases with the carrying capacity of the prey.

These simulations show that even though a type II predator effects the average abundance of the prey population little, the destabilizing effect can be great enough to bring the total population abundance to near zero. We therefore rerank the predators effectiveness as a biocontrol from best to least: type II, type III, and type I.

Note: In this simulation we also plotted stability points to give a more complete showing of behaviour of these three models. We also increased the resolution of K to show more data because it looked cooler.

Intraguild Predation

Assumptions of Intraguild Predation (IGP) between two species and one resource:

1. Species N_1 and N_2 engage in explicit exploitative competition for resource R
2. N_2 consumes N_1
3. N_1 consumes resource R according to the function $f_1(R)$
4. N_2 consumes resource R according to the function $f_2(R)$
5. N_1 is consumed by N_2 according to the function $y(N_1)$
6. N_2 consumes N_1 according to the function $g(y(N_1))$

Implicit System of Equations

$$\frac{dN_1}{dt} = N_1[f_1(R) - N_2y(N_1)]$$

$$\frac{dN_2}{dt} = N_2[f_2(R) + g(y(N_1))]$$

At Equilibrium

$$\hat{N}_1 > 0 \text{ and } \hat{N}_2 > 0 \text{ and } \hat{R}_i > 0$$

$$\frac{dN_1}{dt} = \hat{N}_1[f_1(R) - y(\hat{N}_1)N_2] = 0$$

$$f_1(R) = y(\hat{N}_1)N_2 > 0$$

$$\frac{dN_2}{dt} = \hat{N}_2[f_2(R) + g(y(N_1))] = 0$$

$$f_2(R) = -g(y(N_1)) < 0$$

$$f_2(R) = -g(y(N_1)) < 0 < f_1(R) = y(\hat{N}_1)N_2$$

At Coexistence with IGP

$$f_2(R) < f_1(R)$$

At Coexistence without IGP

$$f_1(R) = f_2(R)$$

IGP is likely to increase coexistence because of the range of parameters and variability that the expression $f_2(R) < f_1(2)$ can maintain. This is in stark contrast to the specific equality of $f_2(R) = f_1(2)$ without intraguild predation. In other words, the model without IGP needs to be numerically equivalent for coexistence to exist whereas the model with IGP can sustain a variety of functional responses as long as $f_2(R) < f_1(2)$.

Explicit System of Equations

$$\frac{dN_1}{dt} = N_1[a_1R - m_1 - bN_2]$$

$$\frac{dN_2}{dt} = N_2[a_2R - m_2 + cbN_1]$$

$$\frac{dR}{dt} = R[(r - \delta) - \frac{a_1}{Y_1}N_1 - \frac{a_2}{Y_2}N_2]$$

At Equilibrium

$$\hat{N}_1 > 0 \text{ and } \hat{N}_2 > 0 \text{ and } \hat{R} > 0$$

$$\frac{dN_1}{dt} = \hat{N}_1[a_1R_1^* - m_1 - bN_2] = 0$$

$$a_1R_1^* - m_1 = bN_2 > 0$$

$$\frac{dN_2}{dt} = \hat{N}_2[a_2R_2^* - m_2 + cbN_1] = 0$$

$$a_2R_2^* - m_2 = -cbN_1 < 0$$

$$a_2R_2^* - m_2 = -cbN_1 < 0 < a_1R_1^* - m_1 = bN_2$$

At Coexistence with IGP

$$a_2R_2^* - m_2 < a_1R_1^* - m_1$$

At Coexistence without IGP

$$R_1^* = R_2^*$$

$$\frac{m_1}{a_1} = \frac{m_2}{a_2}$$

The hypothesis as stated above is still supported even with explicit equations. Coexistence appears to have a wider parameter space in the IGP model than the single equality that $R_1^* = R_2^*$ in the without IGP model.