Assignment # 3: Multispecies Dynamics (Competition and Predation)

In this assignment, you will model multispecies population dynamics.

Part I: Competition

For modeling interspecific competition, we will use the following discrete-time Lotka-Volterra competition equations:

$$N_{1,t+1} = N_{1,t}e^{r1(\frac{K_1 - N_{1,t} - \alpha N_{2,t}}{K_1})}$$
$$N_{2,t+1} = N_{2,t}e^{r1(\frac{K_2 - N_{2,t} - \beta N_{1,t}}{K_2})}$$

In these equations, $N_{i,t}$ and $N_{i,t+1}$ are the population sizes of species **i** at time **t** and **t+1**. The parameter, r_i, is the intrinsic rate of increase of species **i**. The parameter, K_i, is the carrying capacity of species **i**. The terms α and β are per capita competitive effects (competition coefficients) of species 2 on species 1, and species 1 on species 2, respectively.

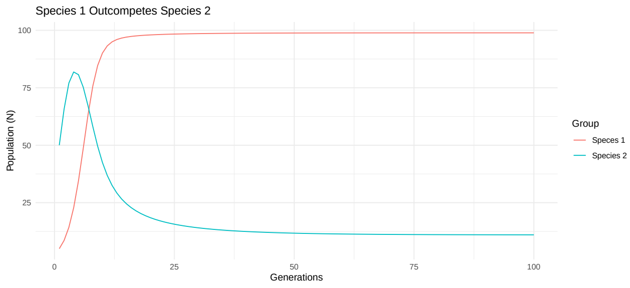
In R, the equations will look something like:

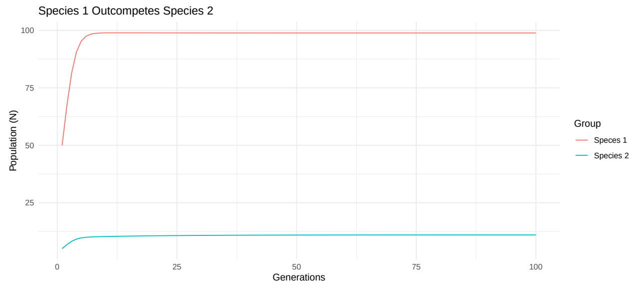
```
N1 = Nt1 * exp(r1*((K1 - Nt1 - alph * Nt2)/K1))
N2 = Nt2 * exp(r2*((K2 - Nt2 - bet * Nt1)/K2))
```

The outcome of competition depends on the relative strengths of intraspecific versus interspecific competition. (Hint: what are the values of the competition coefficients for intraspecific competition?) You need to find combinations of α and β that produce each of the four distinct outcomes of 2-species competition.

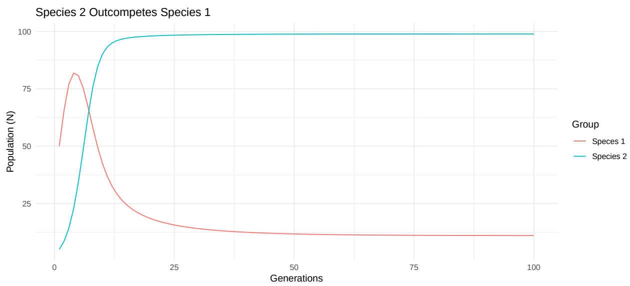
You will need to start each combination of α and β at two different initial population sizes to distinguish between alternate stable states and competitive exclusion. For the two initial conditions, use $N_{1,0} = 5$ & $N_{2,0} = 50$ and $N_{1,0} = 50$ & $N_{2,0} = 5$. Set K_1 and K_2 equal to your carrying capacity from Assignment 1. Pick a value for r_1 and r_2 that is between 0.5 and 1.

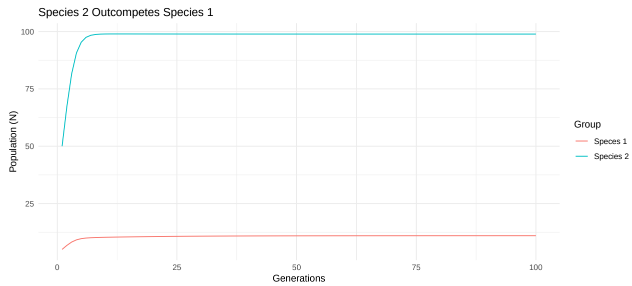
Outcome 1:



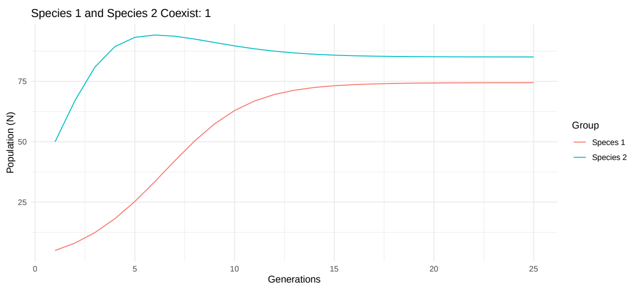


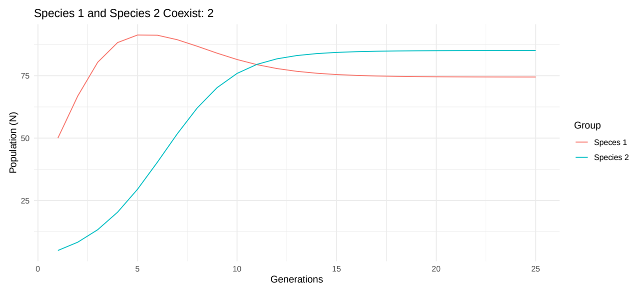
Outcome 2:



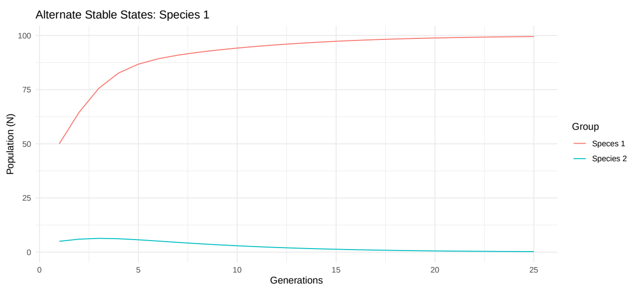


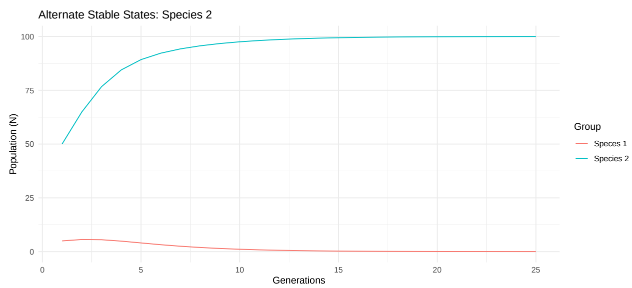
Outcome 3:





Outcome 4:





Part II: Predation

In this exercise, you will model a predator-prey interaction using a discrete-time form of the Rosenzweig-MacArthur predator-prey model.

$$N_{t+1} = N_t e^{r(\frac{K-N_t}{K}) - \frac{aP}{(1+ahN)}}$$

$$P_{t+1} = P_t e^{\frac{baN_t}{(1+ahN_t)} - d}$$

In this model, N_t and P_t are prey and predator populations, respectively. Note that in absence of predator, prey population grows logistically (where parameters r and K are the same as in the logistic model). With predators present, preys suffer additional mortality from predation. Predator growth is a function of prey harvest rate, the conversion efficiency of captured prey into new predators, b, and a density independent per capita death rate, d. The per capita harvest rate of prey by predators is known as a functional response. The details of the functional response are given by a, the encounter rate of predators with prey, and h, the "handling time", that is, time it takes each predator to handle a single prey item.

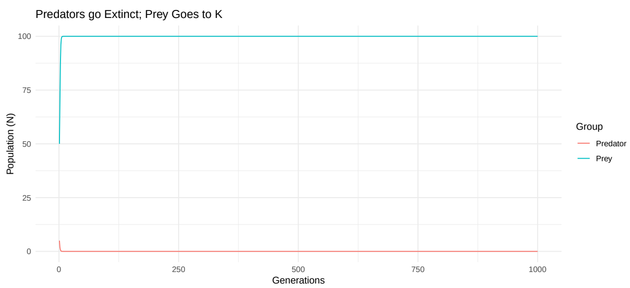
In R, the equations will look something like:

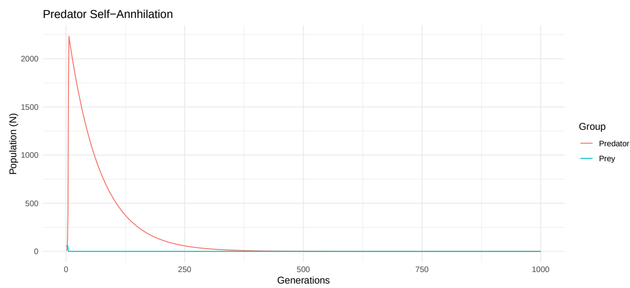
```
pred[i] <- Npred * exp((b*a*Nprey/(1+a*h*Nprey))-d)
prey[i] <- Nprey * exp(r*((K - Nprey)/K) - ((a * Npred)/(1+a*h*Nprey)))
```

Because the stability properties of the system depend on several parameters, changing any one of them can drastically alter the outcome. To make things easier, you are given most of the parameters. Play around with predator death rates d to come up with various outcomes (Hint: how does d affect the isoclines of the system?). Always start your prey population at 50 and your predator population at 5. Choose your K from any number ranging from 95 to 105.

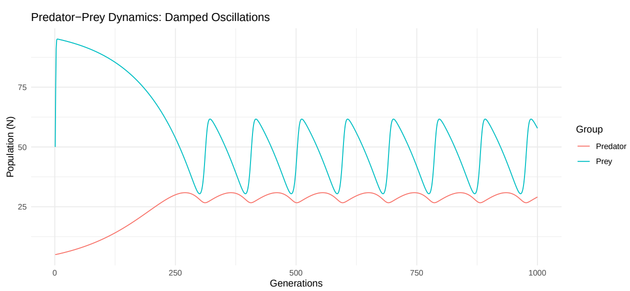
Once our data is generated, our plots will look something like this:

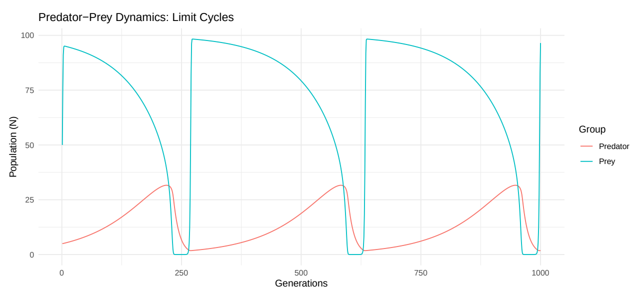
Outcomes 5 and 6 (predator extinction)





Outcomes 7 and 8 (limit cycles and damped oscillations)





Plotting Species 1 and 2 Dynamics

To visualize our data, we will use the following plotting function:

```
plot.growth2 <- function(mydata, mytitle){

  p <- ggplot2::ggplot(mydata, aes(x=Generations, y=value, col=Group)) +
    geom_line() +
    xlab('Generations') +
    ylab('Population (N)') +
    theme_minimal() +
    ggtitle(label = mytitle)

  print(p)

}
```