Introduction to Theoretical Ecology Assignment 8

Stabilizing Lotka-Volterra Predator-Prey Model

In the lab section, we have seen that the basic Lotka-Volterra predator-prey model produces neutral cycles of populations:

$$\frac{dN}{dt} = rN - aNP$$

$$\frac{dP}{dt} = eaNP - \delta P$$

The equations can be modified so that the model can generate stable coexistence of predator and prey.

1. Modify and write down the equations that produce stable coexistence. You can add/change any terms in the original model. (5 pts)

Solution

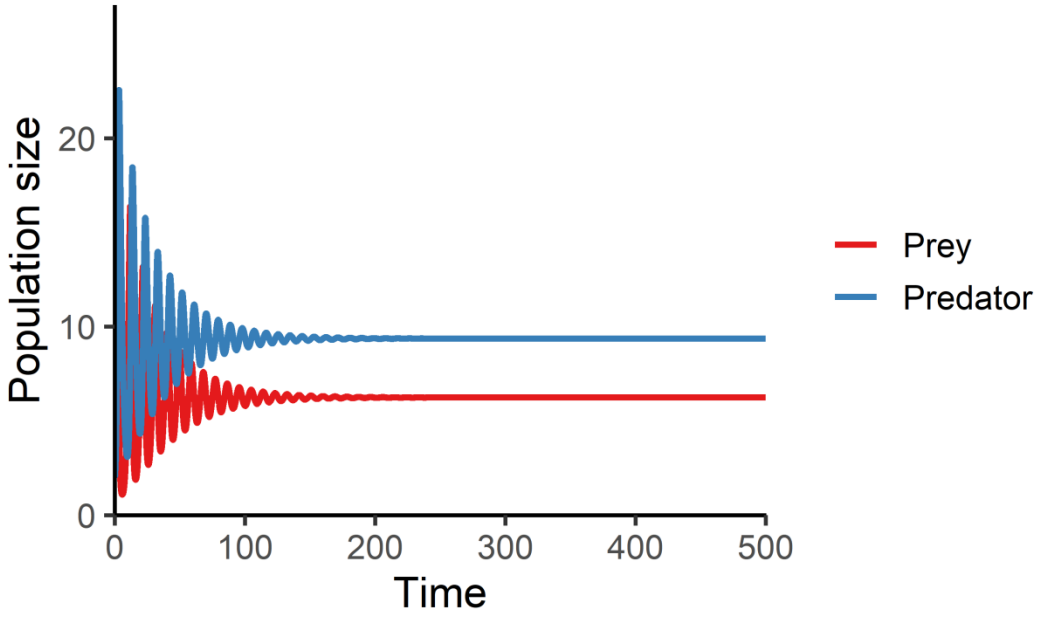
The exponential prey growth in the original model can be changed into logistic growth:

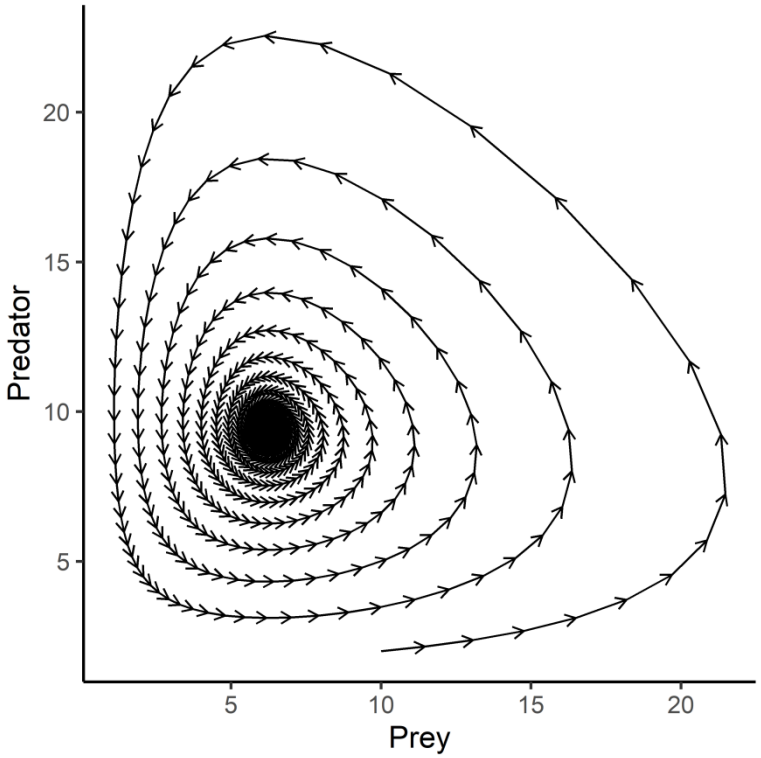
$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right) - aNP$$

$$\frac{dP}{dt} = eaNP - \delta P$$

, where *r* is the intrinsic growth rate of prey, *K* is the carrying capacity of prey, *a* is the capture rate of predator, *e* is the conversion efficiency, and δ is the mortality rate of predator.

2. Select a set of parameters of your choice and visualize the population trajectories demonstrating stable coexistence. (5 pts)





R code

```
library(tidyverse)
library(deSolve)

### Model specification
LV_predation_model <- function(times, state, parms) {
  with(as.list(c(state, parms)), {
    dN_dt = r*N*(1-N/K) - a*N*P
    dP_dt = e*a*N*P - d*P
    return(list(c(dN_dt, dP_dt)))
  })
}

### Model parameters
times <- seq(0, 500, by = 0.01)
state <- c(N = 10, P = 2)
parms <- c(r = 1.0, a = 0.1, e = 0.8, d = 0.5, K = 100)

### Model application
pop_size <- ode(func = LV_predation_model, times = times, y = state,
parms = parms)

### Visualize the population dynamics
# (1) population trajectories
pop_size %>%
  as.data.frame() %>%
  pivot_longer(cols = -time, names_to = "species", values_to = "N")
%>%
```

```
  ggplot(aes(x = time, y = N, color = species)) +
  geom_line(size = 1.5) +
  theme_classic(base_size = 14) +
  labs(x = "Time", y = "Population size") +
  scale_x_continuous(limits = c(0, 500.5), expand = c(0, 0)) +
  scale_y_continuous(limits = c(0, max(pop_size[, -1])*1.2), expand =
c(0, 0)) +
  scale_color_brewer(name = NULL, palette = "Set1", labels = c("Prey",
"Predator"))


# (2) state-space diagram
pop_size %>%
  as.data.frame() %>%
  filter(row_number() %% 20 == 1) %>%
  mutate(N_end = c(N[-1], N[length(N)]),
         P_end = c(P[-1], P[length(P)])) %T>%
  {nr <<- nrow(.)} %>%
  ggplot(aes(x = N, y = P, xend = N_end, yend = P_end)) +
  geom_segment(arrow = arrow(length = unit(seq(0.2, 0, length.out =
nr), "cm"))) +
  theme_classic(base_size = 12) +
  labs(x = "Prey", y = "Predator")
```