Lotka-Volterra Model with Hunting

Marina Kochuten and Jordan Sibley

2025-06-03

GitHub repository: https://github.com/marinakochuten/eds230-hunting

```
# Setup
library(tidyverse)
library(deSolve)
library(here)
library(purrr)

source(here("lotv_mod_hunting.R"))
```

Part 1

Hunting was added to the lotvmodK function in a new R script, `lotv_mod_hunting.R`.

We added hunting by adding a new parameter, **`rhunt`**, which is the rate of hunting, and **`prey_min`**, which is the minimum amount of prey needed before hunting is allowed.

We then say that the number of prey hunted is rhunt * prey if the min_prey is met. Otherwise, number of prey hunted is 0.

We also ensure that our model does not hunt more prey than exist. We do this by choosing whichever number is smaller after calculating hunt - hunt or prey. This means that if hunt is less than prey available, that will be the number of prey hunting. Otherwise, we would hunt all the prey and make hunt equal to prey.

Part 2

Define stability

We are defining stability as maintaining predator and prey populations above 50 after 500 days. I chose these values to allow the population to grow from 1 over time, and because it is a reasonable ratio of predators to prey that also allows the species' to be maintained over time.

```r
# Stability metric function
check_stability <- function(sim_data, prey_thresh = 50, pred_thresh = 50) {
  final_prey <- sim_data %>%
    filter(species == "prey", time == 500) %>%
    pull(population)

  final_pred <- sim_data %>%
    filter(species == "pred", time == 500) %>%
    pull(population)

  # If either value is missing (e.g., model didn't run that far), treat as unstable
  if (length(final_prey) == 0 || length(final_pred) == 0) {
    return(FALSE)
  }

  return(final_prey > prey_thresh && final_pred > pred_thresh)
}
```

Explore hunting

Explore how different hunting levels and different minimum prey populations (before hunting is allowed) are likely to effect the stability of the populations of both predator and prey.

Use this exploration to recommend a hunting target that will be sustainable (e.g leave you with a stable prey and predator population).

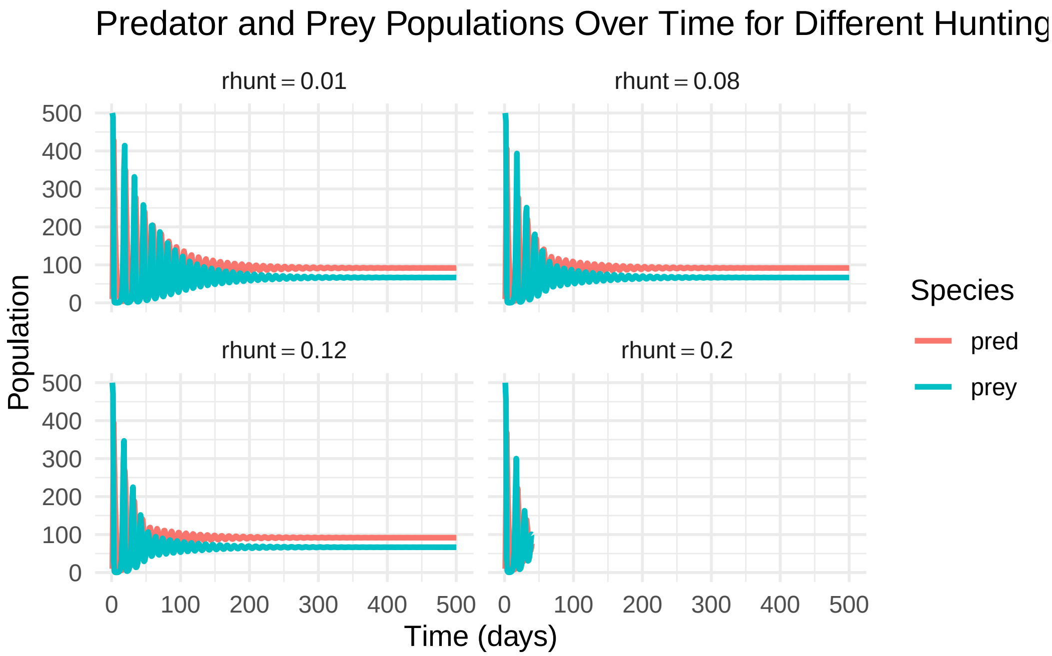
```r
# Set base parameters
base_params <- c(
  rprey = 0.95,
  alpha = 0.01,
  eff = 0.6,
  pmort = 0.4,
  K = 2000
)
```

```r
# Create plot to see species pop over time for various levels of rhunt

# Parameters and setup
rhunt_values <- c(0.01, 0.08, 0.12, 0.2) # vary rhunt values from 0.01 to 0.2
days <- seq(from = 1, to = 500) # set series of days to graph over
```

```r
currpop <- c(prey = 500, pred = 10) # set current prey and predator populations

# Run ODE model for each rhunt
results <- lapply(rhunt_values, function(r) {
  pars <- c(base_params, rhunt = r, prey_min = 100)
  res <- ode(y = currpop, times = days, func = lotv_mod_hunting, parms = pars)
  res_df <- as.data.frame(res)
  colnames(res_df) <- c("time", "prey", "pred")
  res_long <- res_df %>%
    pivot_longer(cols = c(prey, pred), names_to = "species", values_to = "population") %>%
    mutate(rhunt = r)
  return(res_long)
})
```

```
DLSODA-  At current T (=R1), MXSTEP (=I1) steps
      taken on this call before reaching TOUT
In above message, I1 = 5000

In above message, R1 = 39.6318
```

```r
# Combine into one data frame
all_results <- bind_rows(results)

# Create plot for each result of each rhunt value
ggplot(all_results, aes(x = time, y = population, color = species)) +
  geom_line(size = 1) +
  facet_wrap(~ rhunt, labeller = label_bquote(rhunt == .(rhunt))) +
  labs(
    title = "Predator and Prey Populations Over Time for Different Hunting Rates",
    x = "Time (days)",
    y = "Population",
    color = "Species"
  ) +
  theme_minimal()
```



Based on the results of these plots, we will test the models using a hunting rate between 0.01 and 0.2. It seems that anything beyond that range would result in a lot of unstable population levels. Additionally, we will explore the different values of minimum prey population for hunting to be allowed (prey_min) as that is a parameter that is directly related to hunting. From this, we should be able to get a sense of the parameter range that would allow for sustainable hunting.

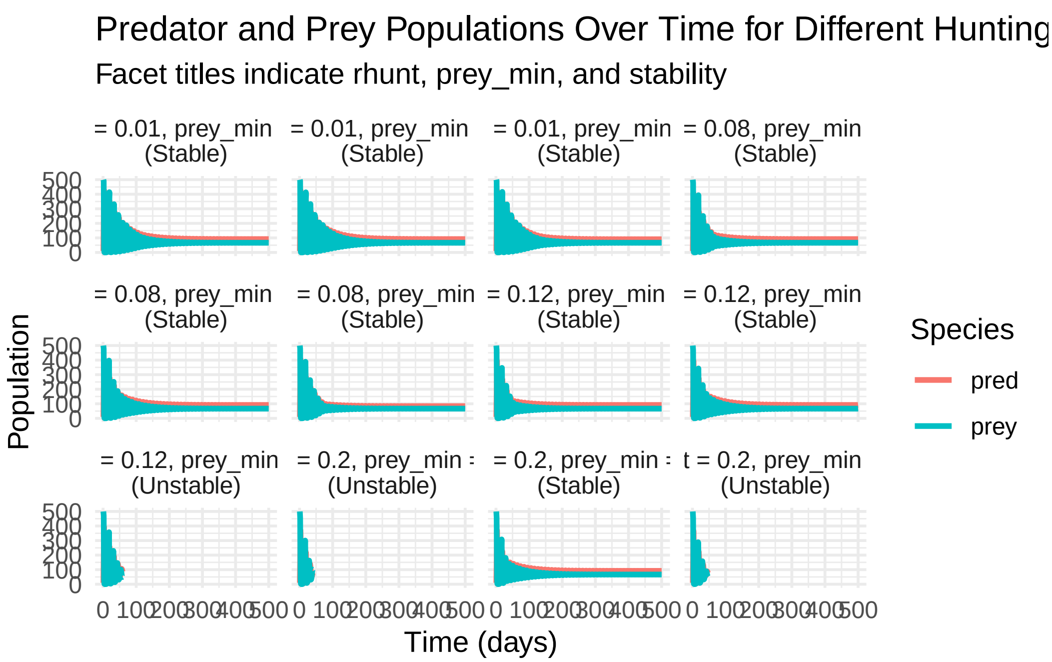
```
# Hunting and prey_min values to test
rhunt_values <- c(0.01, 0.08, 0.12, 0.2)
prey_min_values <- c(50, 100, 150)

# Time and initial populations
days <- seq(from = 1, to = 500)
currpop <- c(prey = 500, pred = 10)

# Create all combinations of rhunt and prey_min
param_grid <- expand.grid(rhunt = rhunt_values, prey_min = prey_min_values)

# Run model and store results
results <- pmap(param_grid, function(rhunt, prey_min) {
  pars <- c(base_params, rhunt = rhunt, prey_min = prey_min)
  res <- ode(y = currpop, times = days, func = lotv_mod_hunting, parms = pars)
  res_df <- as.data.frame(res)
```

```r
  colnames(res_df) <- c("time", "prey", "pred")

  res_long <- res_df %>%
    pivot_longer(cols = c(prey, pred), names_to = "species", values_to = "population") %>%
    mutate(rhunt = rhunt, prey_min = prey_min)

  return(res_long)
})
```

```
DLSODA-  At current T (=R1), MXSTEP (=I1) steps
      taken on this call before reaching TOUT
In above message, I1 = 5000

In above message, R1 = 58.8641

DLSODA-  At current T (=R1), MXSTEP (=I1) steps
      taken on this call before reaching TOUT
In above message, I1 = 5000

In above message, R1 = 46.6221

DLSODA-  At current T (=R1), MXSTEP (=I1) steps
      taken on this call before reaching TOUT
In above message, I1 = 5000

In above message, R1 = 39.6318
```

```r
# Apply stability check
stability_results <- map_lgl(results, check_stability)

# Build stability label table
stability_labels <- param_grid %>%
  mutate(stable = ifelse(stability_results, "Stable", "Unstable"))

# Combine all results
all_results <- bind_rows(results)

# Add stability labels to each simulation
all_results_labeled <- left_join(all_results, stability_labels, by = c("rhunt", "prey_min"))
  mutate(facet_label = paste0("rhunt = ", rhunt, ", prey_min = ", prey_min, "\n(", stable, "
```

```r
# Plot
ggplot(all_results_labeled, aes(x = time, y = population, color = species)) +
  geom_line(size = 1) +
  facet_wrap(~ facet_label) +
  labs(
    title = "Predator and Prey Populations Over Time for Different Hunting Scenarios",
    subtitle = "Facet titles indicate rhunt, prey_min, and stability",
    x = "Time (days)",
    y = "Population",
    color = "Species"
  ) +
  theme_minimal()
```



Results and recommendations

We ran the model across multiple values of the hunting rate of prey as well as with different values of minimum prey population for hunting to be allowed. The values of hunting rate included 0.01, 0.08, 0.12, 0.2, with a larger value indicating a higher hunting pressure. The minimum prey populations for hunting to be allowed were 50, 100, 150. Of the 4 values of hunting rate, all of them besides 0.2 and 0.12 resulted in a stable population of predator and prey regardless of the minimum prey populations for hunting. However, for hunting rate equal to 0.12, it was unstable for a minimum prey population of 50. And the hunting rate value of 0.2

resulted in a unstable status for all prey populations besides when minimum prey population for hunting was equal to 150. In order to maintain stable prey and predator populations while hunting, it is recommended that the hunting rate remain at or below 0.12 and if it is at 0.12, the minimum prey population for hunting to be allowed must exceed 50 individuals.