Implementing the Birth-Death Model for Earthworms

Your Name

Implementing the Birth-Death Model for Earthworms

Introduction

Earthworm populations consist of distinct life stages—cocoons, juveniles, and adults—each with unique survival and reproduction probabilities. To model these dynamics, we implement a birth-death model that integrates life stages and a preferential attachment mechanism. The model allows for a deeper understanding of population dynamics and adaptation under various environmental conditions.

Birth-Death Model with Life Stages

Defining Life Stages

The population is divided into three life stages: 1. **Cocoons (C)**: Represent the egg stage with no reproductive contribution. 2. **Juveniles (J)**: Young earthworms that survive but do not reproduce. 3. **Adults (A)**: Mature earthworms that contribute to reproduction.

The state of the population at time t is represented as:

$$N(t) = \{C(t), J(t), A(t)\},$$

where $C(t)$, $J(t)$, and $A(t)$ denote the number of cocoons, juveniles, and adults, respectively.

Transition Probabilities

Let: - p_c: Probability that a cocoon transitions to a juvenile. - p_j: Probability that a juvenile survives to become an adult. - p_a: Probability that an adult survives.

The transitions are modeled as:

$$C(t+1) = r_a \cdot A(t),$$

$$J(t+1) = p_c \cdot C(t),$$

$$A(t+1) = p_j \cdot J(t) + p_a \cdot A(t),$$

where r_a is the reproduction rate of adults (number of cocoons produced per adult).

Incorporating Preferential Attachment

Reproduction is influenced by fitness categories, with adults in higher fitness categories contributing more cocoons. Let F_i denote the fitness of the i-th adult. The reproduction rate for an adult is proportional to its fitness:

$$R_i = r_a \cdot \frac{F_i}{\sum_j F_j},$$

where the denominator normalizes fitness contributions.

Stochastic Birth-Death Process

The model is implemented as a stochastic process: 1. **Births**: Each adult contributes cocoons based on R_i. 2. **Deaths**: Each individual in $J(t)$ and $A(t)$ survives with probabilities p_j and p_a, respectively.

The population dynamics can be simulated as:

$$P(C(t+1)=k) = \text{Poisson}(\lambda = \sum_i R_i),$$

$$P(J(t+1)=k) = \text{Binomial}(n = C(t), p = p_c),$$
$$P(A(t+1)=k) = \text{Binomial}(n = J(t), p = p_j) + \text{Binomial}(n = A(t), p = p_a).$$

Numerical Simulation

The model is implemented numerically to explore population dynamics over time. Key parameters include: - Initial population sizes $C(0)$, $J(0)$, $A(0)$. - Transition probabilities p_c, p_j, p_a. - Reproduction rate r_a.

Example Simulation in R

```r
# Parameters
initial_population <- list(C = 100, J = 50, A = 20)
transition_probs <- list(p_c = 0.6, p_j = 0.7, p_a = 0.8)
r_a <- 5
n_steps <- 100

# Initialize population
population <- matrix(0, nrow = n_steps, ncol = 3)
colnames(population) <- c("C", "J", "A")
population[1, ] <- unlist(initial_population)

# Simulation
for (t in 2:n_steps) {
  # Reproduction by adults
  new_cocoons <- rpois(1, lambda = r_a * population[t - 1, "A"])

  # Transition to juveniles
  new_juveniles <- rbinom(1, size = population[t - 1, "C"], prob = transition_probs$p_c)

  # Transition to adults
  surviving_adults <- rbinom(1, size = population[t - 1, "A"], prob = transition_probs$p_a)
  new_adults <- rbinom(1, size = population[t - 1, "J"], prob = transition_probs$p_j)

  # Update population
  population[t, "C"] <- new_cocoons
  population[t, "J"] <- new_juveniles
  population[t, "A"] <- new_adults + surviving_adults
}
```

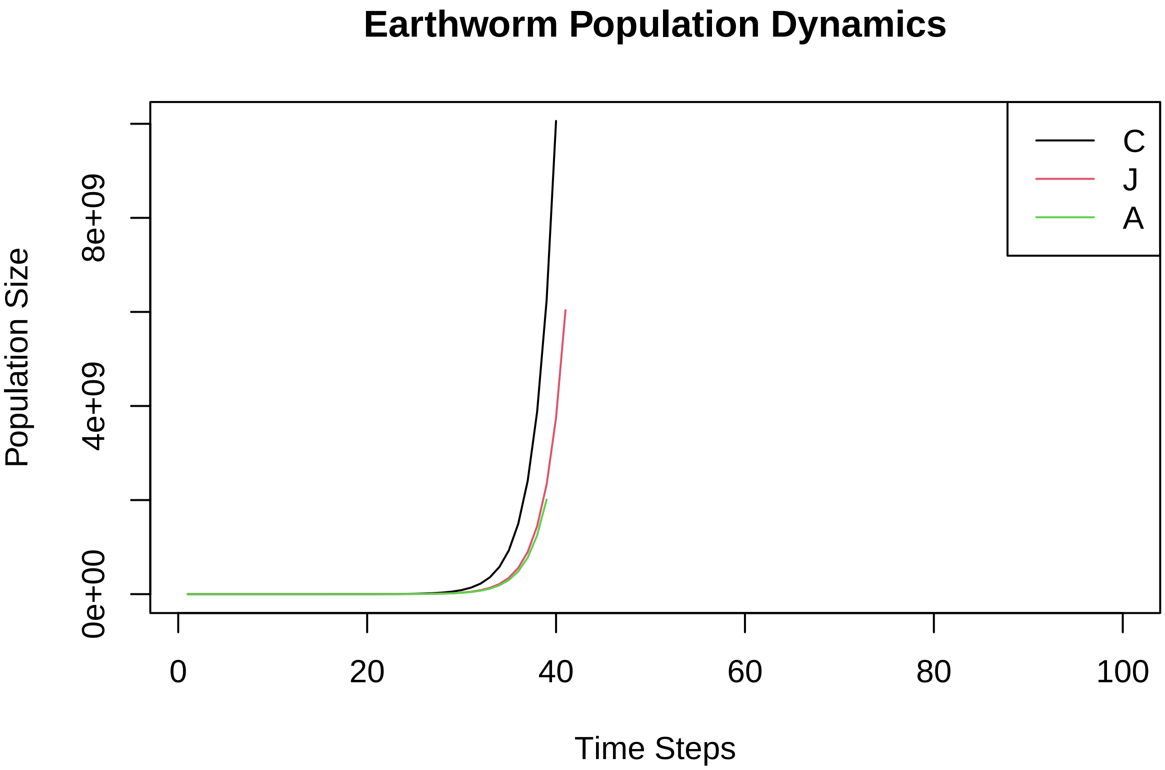
```
## Warning in new_adults + surviving_adults: NAs produced by integer overflow
```

```
## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced
```

```
## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced
```

```
## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced
## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced
## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced
## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced
## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced
## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced
## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced
## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced
## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced
## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced
## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced

## Warning in rpois(1, lambda = r_a * population[t - 1, "A"]): NAs produced

## Warning in rbinom(1, size = population[t - 1, "C"], prob =
## transition_probs$p_c): NAs produced

## Warning in rbinom(1, size = population[t - 1, "A"], prob =
## transition_probs$p_a): NAs produced

## Warning in rbinom(1, size = population[t - 1, "J"], prob =
## transition_probs$p_j): NAs produced
```

```r
# Plot results
matplot(1:n_steps, population, type = "l", lty = 1, col = 1:3,
        xlab = "Time Steps", ylab = "Population Size",
        main = "Earthworm Population Dynamics")
legend("topright", legend = colnames(population), col = 1:3, lty = 1)
```



Applications and Implications

Environmental Stressors

The model can predict how environmental changes (e.g., pollution) impact survival and reproduction probabilities, altering population dynamics.

Conservation Efforts

Understanding life-stage-specific dynamics can inform strategies for conserving earthworm populations, such as enhancing survival rates of juveniles.

Vermicomposting Optimization

The preferential attachment mechanism highlights the importance of maintaining a healthy population of high-fitness adults for optimal reproduction.

Conclusion

The birth-death model provides a robust framework for analyzing earthworm population dynamics. By incorporating life stages and fitness-based reproduction, it captures key ecological processes critical for managing and conserving earthworm populations. “‘