Vole Simulations

Elizabeth Hiroyasu

April 10, 2017

To look at the sensitivity of the prey population to the model parameters, we can run the model for a variety of different parameters and look how the curve changes. For this simulation, we will set a variety of r values for population runs for vole populations.

Population growth rates

From the literature review (Sarah this is the average data from the model summary data), we constructed minimum, average, and maximum vole stage structured matrices. It is important to note that survival measurements are from M. townsendii and fertility measurements were from M. californicus because no studies on either species contained both measurements.

```
##          Juvenile   Adult
## Juvenile     0.69 0.07935
## Adult        0.20 0.02300
```

```
##          Juvenile Adult
## Juvenile   1.4056 1.757
## Adult      0.2800 0.350
```

```
##          Juvenile  Adult
## Juvenile   2.6896 3.3456
## Adult      0.4100 0.5100
```

Because the calculated r-values for voles is for a 10 week period, and the predator model is in a time step of seasons, we have to convert the r-value to a seasonal r. To provide some context, annual r-values are also calculated.

```
r_annual<-r_vole*5
r_annual
```

```
## [1] -1.7  2.8  5.8
```

```
if(r_annual[1] <0){
  r_annual[1]=0.04
}

r_vole<-r_annual/4
r_vole
```

```
## [1] 0.01 0.70 1.45
```

Parameters

We can use the pred_prey function created in our owls r-package, then use an ode solver to solve the derivatives. We will use a type two functional response to model the behavior of the prey population. A type two functional response is the simplest model to use in the wake of low data. The response will capture typical more realistic predator-prey relations. The differental equations are below:

$$\frac{dprey}{dt} = rN(1 - \frac{N}{K_{prey}}) - \frac{k_{max}N}{N+D}P$$
$$\frac{dpredator}{dt} = \beta PN - \delta P$$

The model requires multiple parameters: r = growth rate of prey pop (voles/season) This value is extracted from the population matrices using the popbio package to calculate lambda, then r=ln(lambda)

K_prey = carrying capacity of prey (voles)

alpha = attack rate of predator (or capture efficiency; the larger alpha is, the more the per capita growth rate of the prey population is depressed by the addition of a single predator) (units=1/season) The attack rate was calculated from Derting and Cranford (1989). They found that, on average, it takes 1.1 attacks/vole, and the probability of capture was 0.84 for microtus adults (number of successful predator attacks/total number of attacks).

h = handling time The amount of time it takes an owl to kill and eat a single vole (1/k_max) (season/vole)

beta= assimilation efficiency (efficiency of turning voles into per capita growth) This number is calculated from field data and represents the fractional value of one vole to an owl producing 4.33 chicks.

delta = death rate of predator (#predator/season)

k_max = maximum feeding rate the most gophers an owl can eat in a single time unit (voles/season). Calculated from empirical data.

D = half saturation constant (1/(alpha*handling time)) The abundance of prey at which the feeding rate is half maximal (gopher)

State variables: N = starting population of prey P = starting population of predator

For this model, time is in terms of season (3 months), so each time interval is represented by a season and four seasons are equivalent to one year.

```r
#Parameters
alpha=3.0
beta = 5.85e-4
delta=0.01
K_prey=1000
k_max=654
h=1/k_max
D=1/(alpha*h)
r=r_vole
parameters <- c(r, alpha, beta, delta, K_prey, k_max, D)


#State variables:
N=c(K_prey, K_prey, K_prey,
    0.5*K_prey, 0.5*K_prey, 0.5*K_prey,
    2,2,2)

P=c(0.2, 0.6, 1.0,
    0.2, 0.6, 1.0,
    0.2, 0.6, 1.0)

state<-cbind(N, P)
times<- seq(0, 20, by=1)
```

High Simulations, K=1000, alpha= 3

We can then run simulations for these various lists of r values while varying predator density.

Then, plotting the prey density over time by each N.

Table 1: N init= 1000

time	N	P	r	K	alpha
20	0.7667699	0.2	0.01	1000	3
20	0.0000000	0.6	0.01	1000	3
20	0.0000000	1.0	0.01	1000	3
20	819.9908012	0.2	0.70	1000	3
20	0.0004257	0.6	0.70	1000	3
20	0.0000000	1.0	0.70	1000	3
20	920.7868471	0.2	1.45	1000	3
20	707.6392415	0.6	1.45	1000	3
20	0.0000011	1.0	1.45	1000	3

Table 2: N init= 500

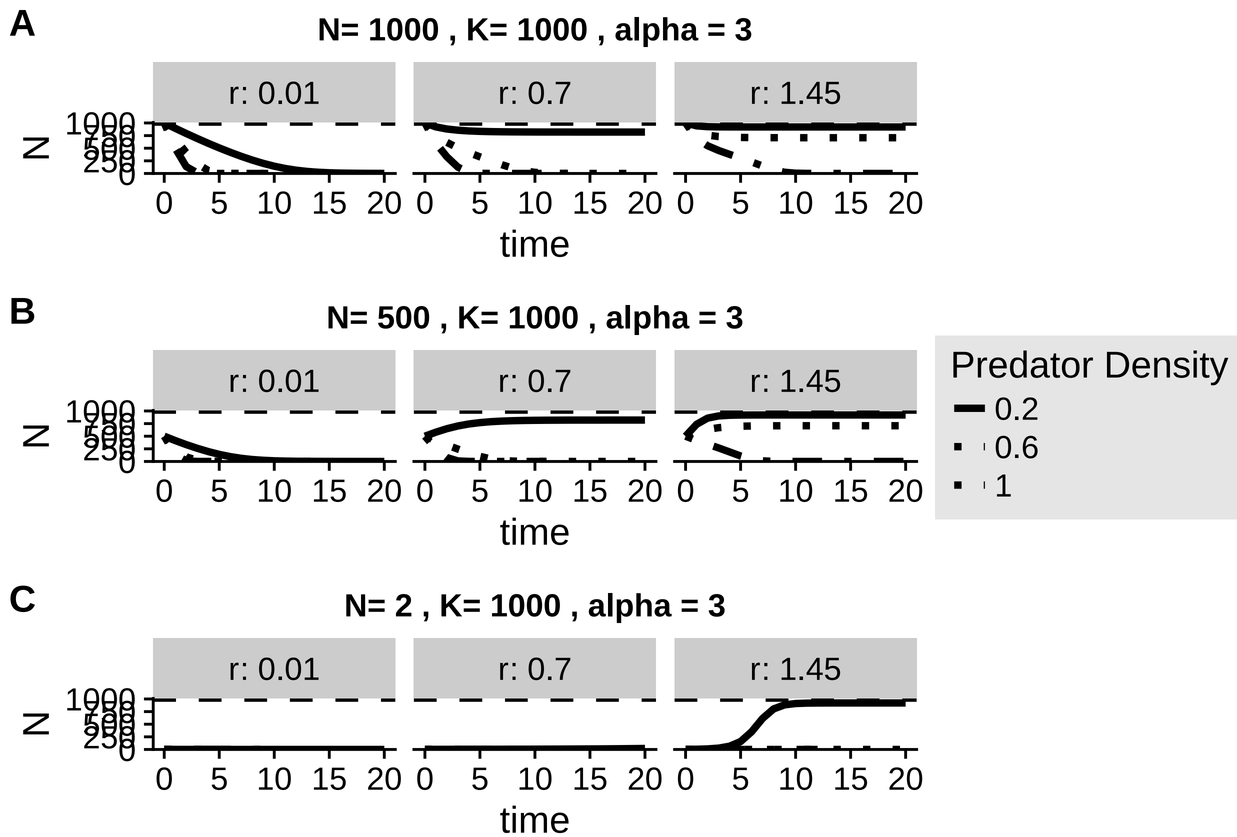
time	N	P	r	K	alpha
20	0.0387372	0.2	0.01	1000	3
20	0.0000000	0.6	0.01	1000	3
20	0.0000000	1.0	0.01	1000	3
20	819.9358505	0.2	0.70	1000	3
20	0.0000148	0.6	0.70	1000	3
20	0.0000000	1.0	0.70	1000	3
20	920.7868474	0.2	1.45	1000	3
20	707.6388061	0.6	1.45	1000	3
20	0.0000000	1.0	1.45	1000	3

Table 3: N init= 2

time	N	P	r	K	alpha
20	0.0000153	0.2	0.01	1000	3
20	0.0000000	0.6	0.01	1000	3
20	0.0000000	1.0	0.01	1000	3
20	19.6250895	0.2	0.70	1000	3
20	0.0000000	0.6	0.70	1000	3
20	0.0000000	1.0	0.70	1000	3
20	920.7867965	0.2	1.45	1000	3
20	0.0018971	0.6	1.45	1000	3
20	0.0000000	1.0	1.45	1000	3

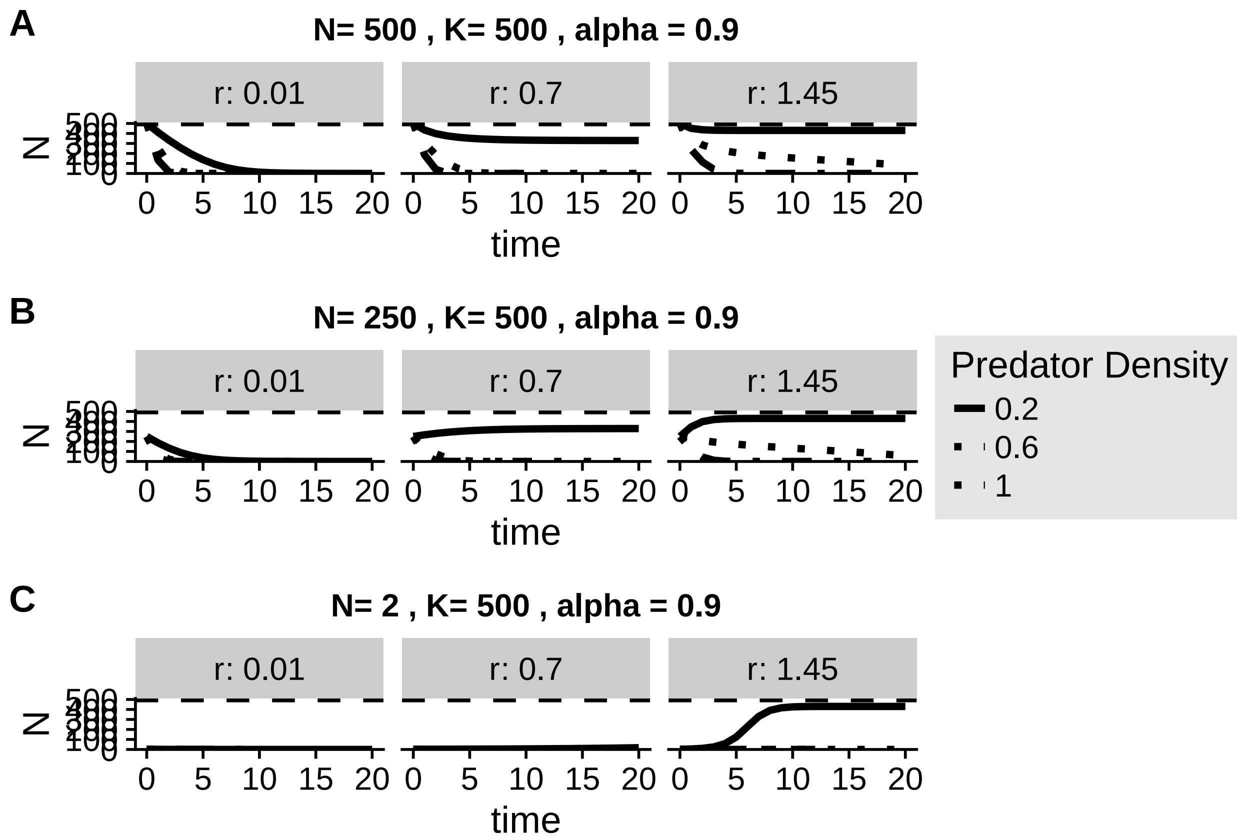
In black and white:

```
## Warning: Removed 8 rows containing missing values (geom_path).
```

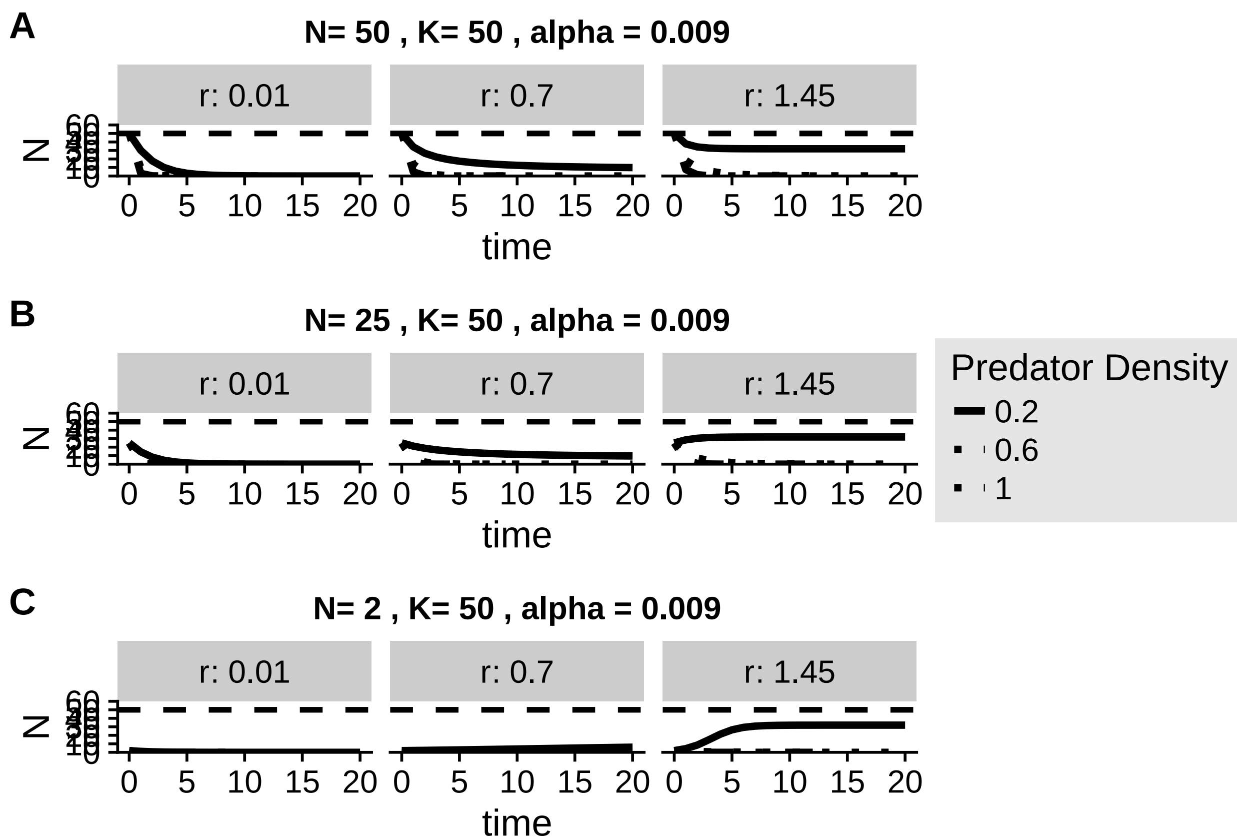


Medium Simulations, K=500, alpha=0.9

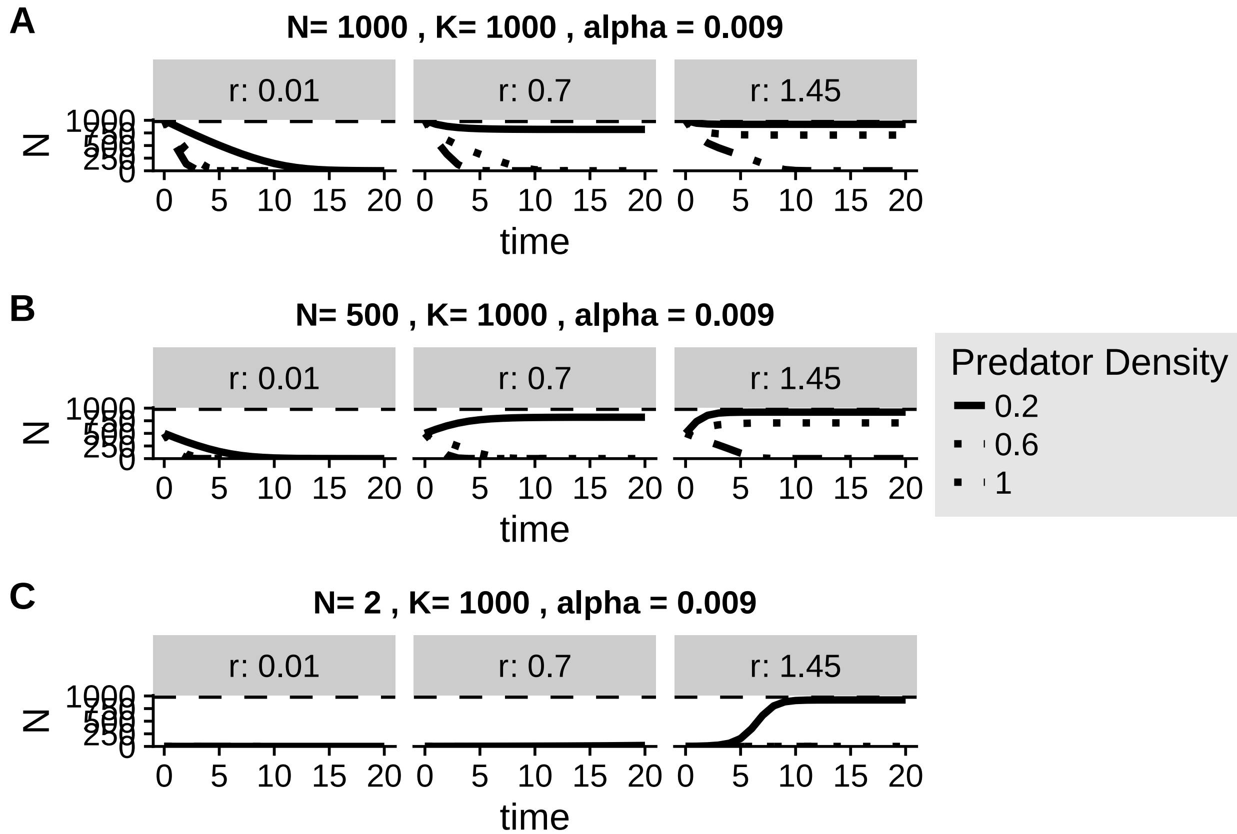
Suppose we want to investigate what happens when the carrying capacity is lower, K=500


A
N= 500 , K= 500 , alpha = 0.9
r: 0.01
r: 0.7
r: 1.45
N
time
B
N= 250 , K= 500 , alpha = 0.9
Predator Density
0.2
0.6
1
C
N= 2 , K= 500 , alpha = 0.9

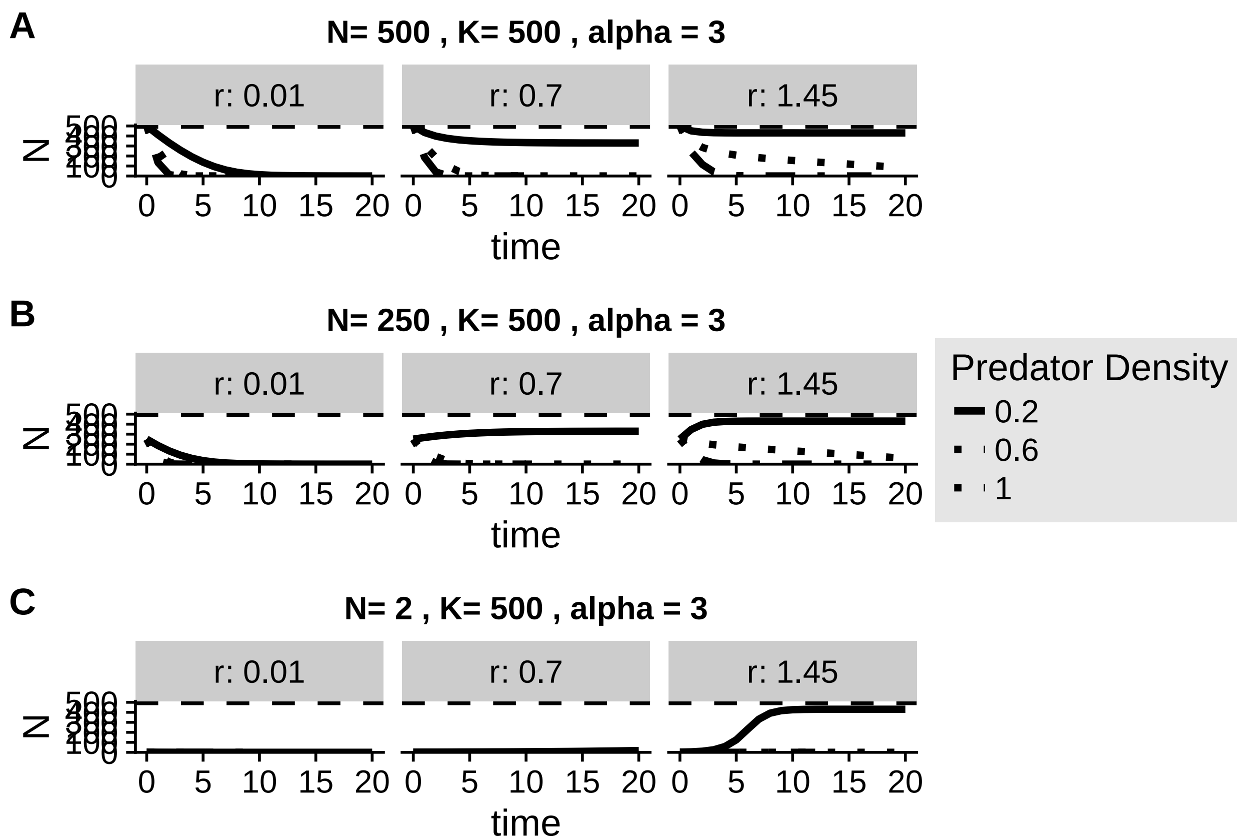
Low Simulations, K=50 alpha= 0.009



Supplemental Simulations, K=1000



Supplemental Simulations, K=500



Supplemental Simulations, K=50