Simulations Varying Gopher Population Carrying Capacity

Elizabeth Hiroyasu

November 22, 2016

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.

Population growth rates

From literature review, we have constructed estimates of the minimum, maximum, and average population matrices of gopher populations. We can use these population matrices to calculate the instantenous growth rate of the population, or the r-value.

```
##          Juvenile SubAdult   Adult
## Juvenile      0.0     0.00 0.12675
## SubAdult      0.5     0.00 0.00000
## Adult         0.0     0.66 0.75000
```

```
##          Juvenile SubAdult Adult
## Juvenile      0.0     0.00 2.376
## SubAdult      0.5     0.00 0.000
## Adult         0.0     0.77 0.800
```

```
##          Juvenile SubAdult  Adult
## Juvenile      0.0     0.00 3.7018
## SubAdult      0.5     0.00 0.0000
## Adult         0.0     0.87 0.8300
```

For clarity, we can calculate the annual growth rate:

```
## [1] 0.04 1.12 1.68
```

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 (gophers/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 (gophers)

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 Crawford (1989). They found that, on average, it takes 1.1 attacks/vole - for this analysis, we assume the attack rate is the same for gophers as for voles. If there are 11 successful attacks per night the average number observed in observations recorded in Bunn et al (1982), then there should be 12.1 total attacks per night. If there is one foraging event per night, and 90 foraging events per season, then we expect 1089 attacks per season. However, this is if there are a significant amount of gophers across the landscape (scaling up from the Derting and Crawford densities, we get over 3000 voles/ha, an unreasonable number). Therefore, we would expect attack rate to decrease with density of the prey population. In sensitivity analysis if the attack rate is less than three, there is coexistence between gopher and owl populations.

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

beta= assimilation efficiency (efficiency of turning gophers into per capita growth) This number is calculated from field data and represents the fractional value of one gopher 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 (gophers/season)

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=0.924
beta = 1.01e-3
delta=0.01
K_prey=175
k_max=378
h=1/k_max
D=1/(alpha*h)
r=r_gopher
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)
```

Simulations, K=175

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= 175

time	N	P	r	K
20	7.5803620	0.2	0.01	175
20	0.0049267	0.6	0.01	175
20	0.0000031	1.0	0.01	175
20	84.1519878	0.2	0.28	175
20	0.5622610	0.6	0.28	175
20	0.0004822	1.0	0.28	175
20	115.1791977	0.2	0.42	175
20	4.8081930	0.6	0.42	175
20	0.0062554	1.0	0.42	175

Table 2: N init= 87.5

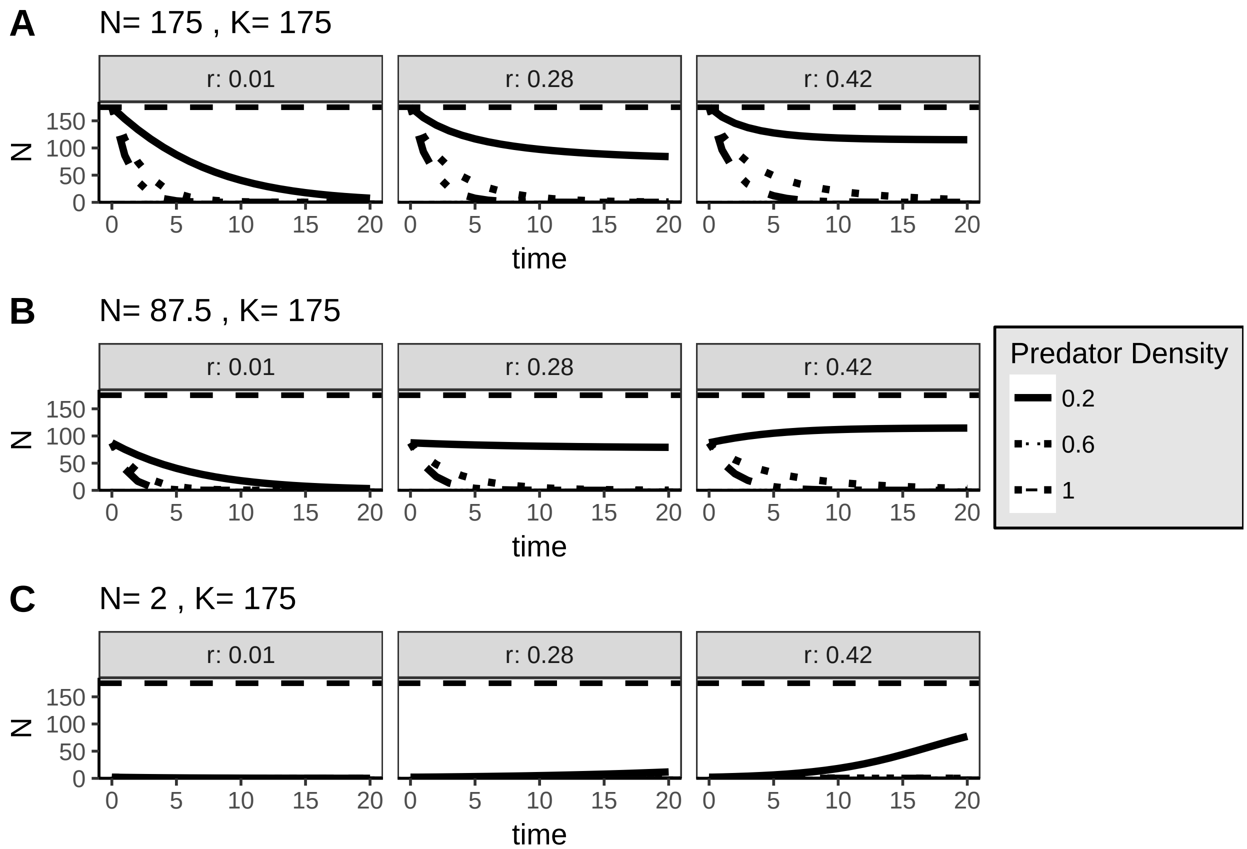
time	N	P	r	K
20	3.1929668	0.2	0.01	175
20	0.0020105	0.6	0.01	175
20	0.0000012	1.0	0.01	175
20	79.1550180	0.2	0.28	175
20	0.3241000	0.6	0.28	175
20	0.0002372	1.0	0.28	175
20	114.6082690	0.2	0.42	175
20	3.4979898	0.6	0.42	175
20	0.0034581	1.0	0.42	175

Table 3: N init= 2

time	N	P	r	K
20	0.0609030	0.2	0.01	175
20	0.0000376	0.6	0.01	175
20	0.0000000	1.0	0.01	175
20	11.7897626	0.2	0.28	175
20	0.0082574	0.6	0.28	175
20	0.0000050	1.0	0.28	175
20	77.2152995	0.2	0.42	175
20	0.1340844	0.6	0.42	175
20	0.0000838	1.0	0.42	175

Plotting in black and white:

```
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
```




```
## Saving 6.5 x 4.5 in image
```

Simulations, K=100

Suppose we want to investigate what happens when the carrying capacity is lower, K=100

Table 4: N init= 175

time	N	P	r	K
20	7.1669726	0.2	0.01	100
20	0.0048294	0.6	0.01	100
20	0.0000030	1.0	0.01	100
20	44.7817834	0.2	0.28	100
20	0.3564867	0.6	0.28	100
20	0.0003589	1.0	0.28	100
20	62.1102731	0.2	0.42	100
20	2.5815394	0.6	0.42	100
20	0.0041168	1.0	0.42	100

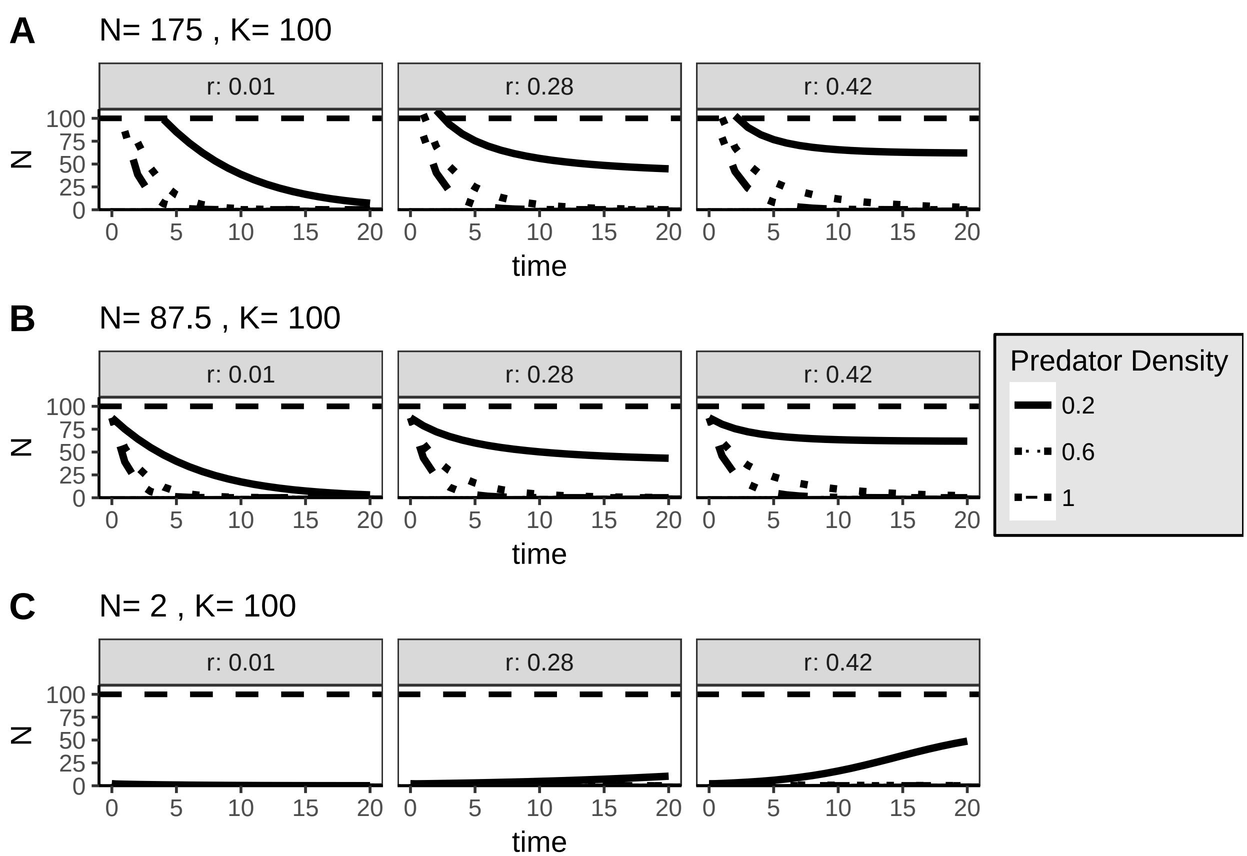
Table 5: N init= 87.5

time	N	P	r	K
20	3.1155695	0.2	0.01	100

time	N	P	r	K
20	0.0019937	0.6	0.01	100
20	0.0000012	1.0	0.01	100
20	43.2348049	0.2	0.28	100
20	0.2422320	0.6	0.28	100
20	0.0002025	1.0	0.28	100
20	61.9265289	0.2	0.42	100
20	2.1398779	0.6	0.42	100
20	0.0026767	1.0	0.42	100

Table 6: N init= 2

time	N	P	r	K
20	0.0608739	0.2	0.01	100
20	0.0000376	0.6	0.01	100
20	0.0000000	1.0	0.01	100
20	10.4665786	0.2	0.28	100
20	0.0081863	0.6	0.28	100
20	0.0000050	1.0	0.28	100
20	48.8377649	0.2	0.42	100
20	0.1308645	0.6	0.42	100
20	0.0000832	1.0	0.42	100



Simulations, K=50

Changing K to 50 **doesn't work if r is negative

Table 7: N init= 175

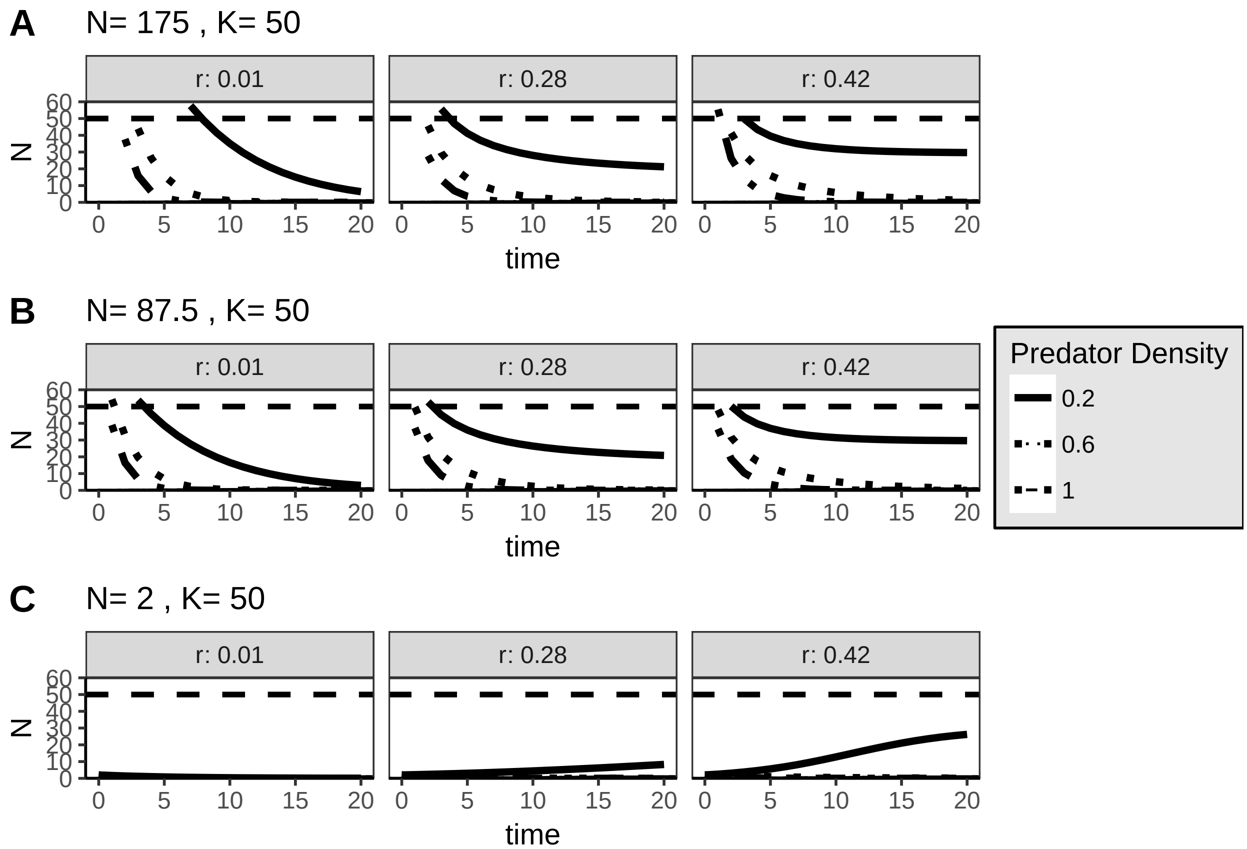
time	N	P	r	K
20	6.3563155	0.2	0.01	50
20	0.0046163	0.6	0.01	50
20	0.0000029	1.0	0.01	50
20	21.2120358	0.2	0.28	50
20	0.1912982	0.6	0.28	50
20	0.0002242	1.0	0.28	50
20	29.6976118	0.2	0.42	50
20	1.2324494	0.6	0.42	50
20	0.0022792	1.0	0.42	50

Table 8: N init= 87.5

time	N	P	r	K
20	2.9487694	0.2	0.01	50
20	0.0019557	0.6	0.01	50
20	0.0000012	1.0	0.01	50
20	20.8456598	0.2	0.28	50
20	0.1522232	0.6	0.28	50
20	0.0001507	1.0	0.28	50
20	29.6533394	0.2	0.42	50
20	1.1197359	0.6	0.42	50
20	0.0017509	1.0	0.42	50

Table 9: N init= 2

time	N	P	r	K
20	0.0608062	0.2	0.01	50
20	0.0000376	0.6	0.01	50
20	0.0000000	1.0	0.01	50
20	8.2944068	0.2	0.28	50
20	0.0080247	0.6	0.28	50
20	0.0000049	1.0	0.28	50
20	26.2382879	0.2	0.42	50
20	0.1239205	0.6	0.42	50
20	0.0000817	1.0	0.42	50



Simulations, K=25

*doesn't work if r is negative

Table 10: N init= 175

time	N	P	r	K
20	5.1799296	0.2	0.01	25
20	0.0042417	0.6	0.01	25
20	0.0000029	1.0	0.01	25
20	10.3007386	0.2	0.28	25
20	0.0989262	0.6	0.28	25
20	0.0001276	1.0	0.28	25
20	14.4931336	0.2	0.42	25
20	0.6007260	0.6	0.42	25
20	0.0011998	1.0	0.42	25

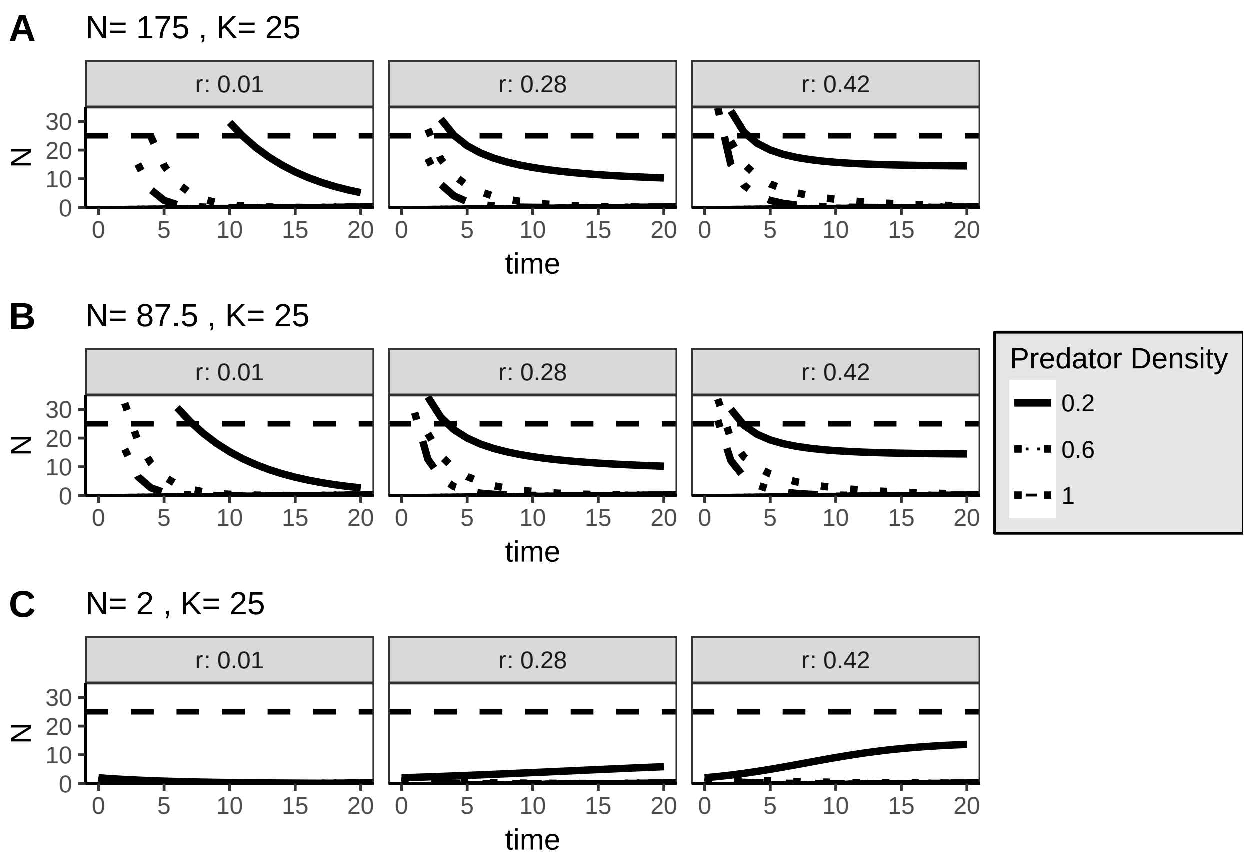
Table 11: N init= 87.5

time	N	P	r	K
20	2.6634003	0.2	0.01	25
20	0.0018838	0.6	0.01	25
20	0.0000012	1.0	0.01	25

time	N	P	r	K
20	10.2122459	0.2	0.28	25
20	0.0872195	0.6	0.28	25
20	0.0000996	1.0	0.28	25
20	14.4823877	0.2	0.42	25
20	0.5722868	0.6	0.42	25
20	0.0010339	1.0	0.42	25

Table 12: N init= 2

time	N	P	r	K
20	0.0606712	0.2	0.01	25
20	0.0000375	0.6	0.01	25
20	0.0000000	1.0	0.01	25
20	5.8612441	0.2	0.28	25
20	0.0077203	0.6	0.28	25
20	0.0000049	1.0	0.28	25
20	13.6124025	0.2	0.42	25
20	0.1120313	0.6	0.42	25
20	0.0000792	1.0	0.42	25



These model simulations show that at the two lowest barn owl densities (0.2 and 0.4/ha), the pocket gopher population is never driven to zero, no matter what the growth rate of the gopher population or carrying capacity. At the lowest barn owl density (0.2 barn owls/ha), the pocket gopher populations are able to grow to carrying capacity. On the opposite end of the spectrum, at the two highest barn owl densities (0.95 and 2/ha) the pocket gopher population is drive to extinction. At the highest barn owl density (2barn owls/ha), this is usually achieved by the middle of the second year. It is important to note that these simulation exercises do not include prey switching, so barn owls may not be able to drive pocket gopher populations extinct, but instead switch prey species when they become harder to find.

The simulations also show that the presence of barn owls, no matter what the initial density and the initial density, growth rate, and carrying capacity, prevent the prey population from reaching carrying capacity. Depending on the growth rate and initial density, the presence of barn owls can cause the equilibrium pocket gopher population size to be much lower than carrying capacity. Therefore if the carrying capacity of the landscape is low, then barn owls may be effective at driving down pocket gopher populations to a manageable level.