Modelling_Group_Project_R

MW, KV, LA, SD

November 27, 2017

Modeling Group Project Part 1 - Lotka-Volterra Model

Model Description

The Lotka-Volterra model is one of the simplest models that describes predator-prey dynamics. Its simplicity comes from assumptions that the only limiting factor for prey population growth is the predator and the only limiting factor for predator population is the availability of prey. It also assumes that these interactions between predator and prey are linear.

Model Equations

Prey $\frac{dh}{dt} = bH - aPH$

Predator $\frac{dP}{dt} = eaPH - sP$

Where H is the prey (herbivore), the predator is P, and b, a, e, and s are prey birthrate, predator attack rate, conversion efficiency of prey to predators, and predator death rate, respectively.

Initial Model

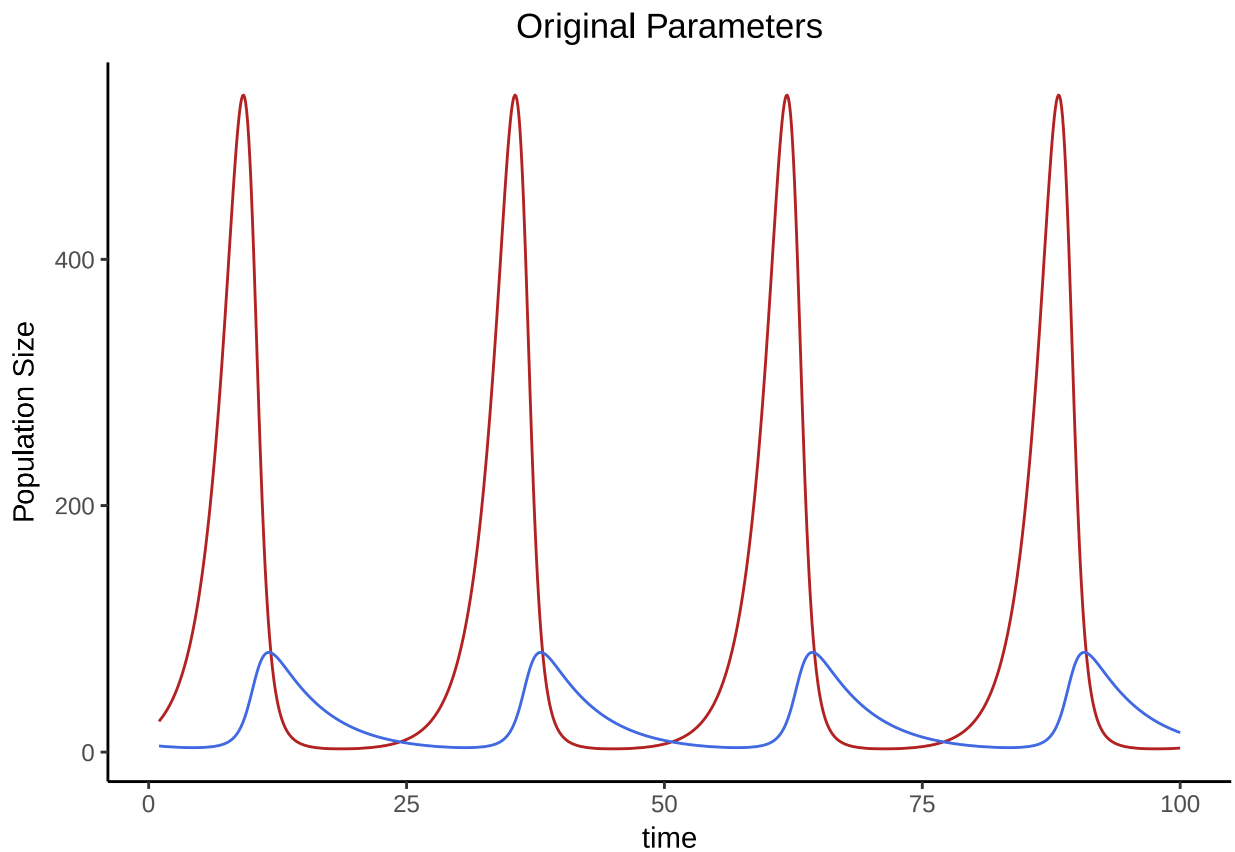
Below we load the required modules, set initial parameters, and use the function ode() to run our model.

```
# Load the deSolve package required for solving differential equations in R and ggplot2 package for gra
library(deSolve)
```

```
## Warning: package 'deSolve' was built under R version 3.4.1
```

```
library(ggplot2)
library(knitr)

# Create a custom function for simulating Lotka-Volterra Competition Model
predPreySim<-function(t,y,p){
  herbivore=y[1]
  carnivore=y[2]
  h_birthrate=p[1]
  c_attackrate=p[2]
  h_c_efficiency=p[3]
  c_death=p[4]
  dHerb_dt=(h_birthrate*herbivore)-(c_attackrate*carnivore*herbivore)
  dCarn_dt=(h_c_efficiency*c_attackrate*carnivore*herbivore)-(c_death*carnivore)
  return(list(c(dHerb_dt,dCarn_dt)))
}


# Set parameters, initial state variables, and simulation timepoints
y = c(25,5)
p = c(0.5,0.02,0.1,0.2)
```

```r
times=seq(from=1,to=100,by=0.1)


# Simulate the model using ode
modelSim=ode(y=y,times=times,func=predPreySim,parms=p)
colnames(modelSim)[2] <- "herbivore"
colnames(modelSim)[3] <- "predator"

# Convert to datafrome for plotting purposes
modelSim2 <- as.data.frame(modelSim)

# plot simulation
ggplot(data=modelSim2,aes(x=time)) + geom_line(aes(y=herbivore), color="firebrick") + geom_line(aes(y=p
```



For all graphs, red is prey and blue is predator. This is true for all graphs to follow.

Initial summary of output below. How the parameters change will be determined by comparing results to above graph and summary below.

```r
#show result summary of the data frame
kable(summary(modelSim))
```

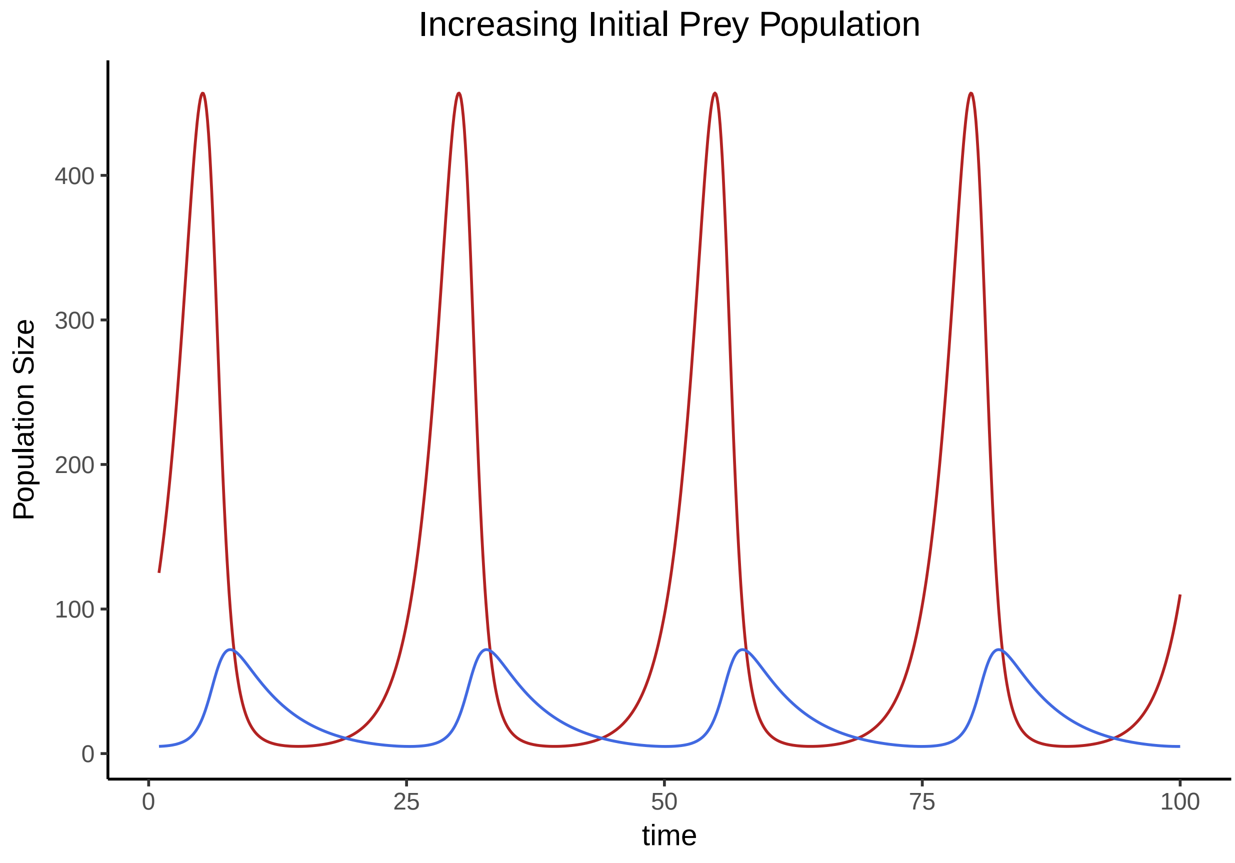
	herbivore	predator
Min.	2.642454	3.676633
1st Qu.	4.364783	5.386827
Median	21.810086	15.707261
Mean	105.735381	26.002544

	herbivore	predator
3rd Qu.	141.380539	41.663795
Max.	533.347326	80.982968
N	991.000000	991.000000
sd	155.689454	24.543401

Changing the number of prey

Adjust initial parameters. Change prey from 25 to 125.

```
# simulate dynamics by resetting parameters and intitial conditions
y = c(125,5)
p = c(0.5,0.02,0.1,0.2)
times=seq(from=1,to=100,by=0.1)

# Simulate the model using ode
modelSim=ode(y=y,times=times,func=predPreySim,parms=p)
colnames(modelSim)[2] <- "herbivore"
colnames(modelSim)[3] <- "predator"

# save model as a data frame for plotting purposes
modelSim2 <- as.data.frame(modelSim)

# plot simulation
ggplot(data=modelSim2) + geom_line(aes(x=time,y=herbivore),color="firebrick")+geom_line(aes(x=time,y=pre
```

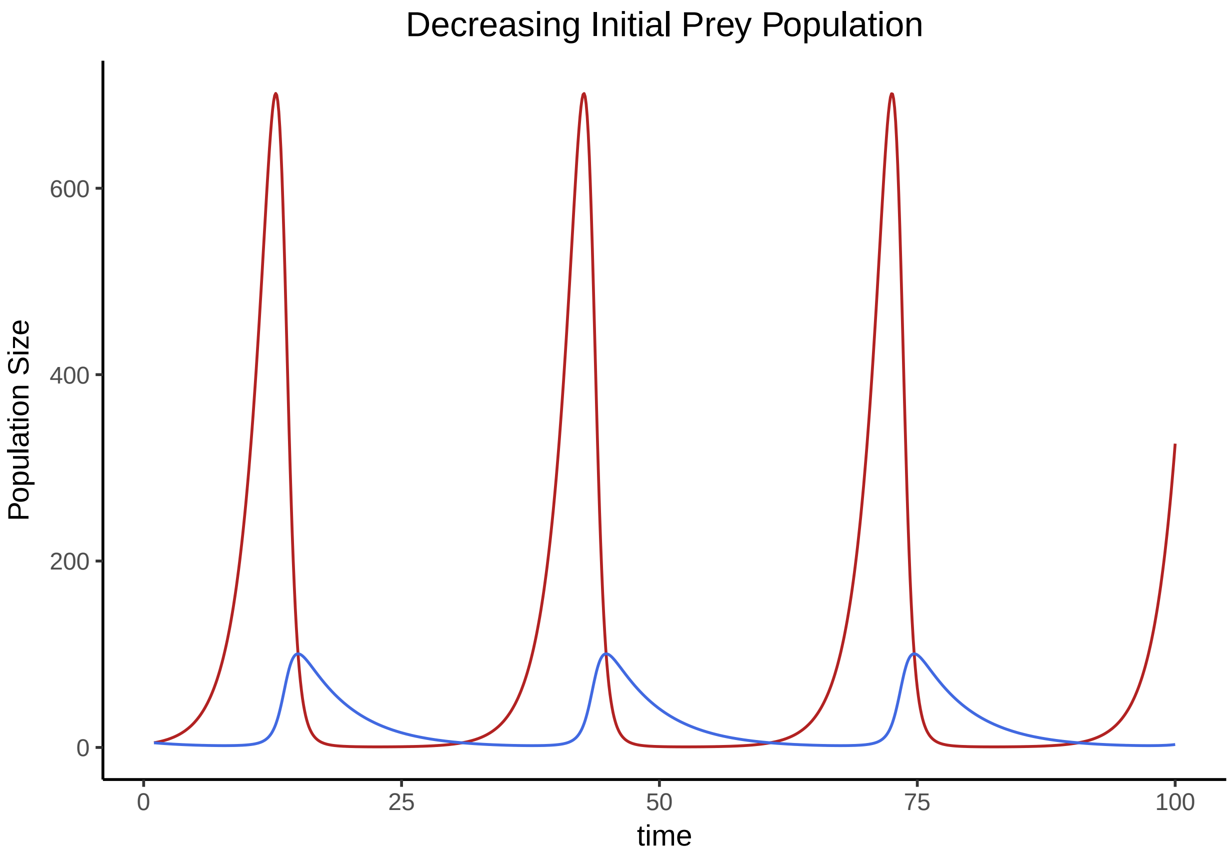


```r
#show result summary of the data frame
kable(summary(modelSim))
```

	herbivore	predator
Min.	4.975246	4.933426
1st Qu.	7.771253	7.059210
Median	26.336462	15.703783
Mean	99.961990	25.043921
3rd Qu.	142.406059	39.363238
Max.	456.995062	71.922218
N	991.000000	991.000000
sd	135.983485	21.480734

Adjust initial parameters again. Change prey from 25 to 5.

```r
# simulate dynamics by resetting parameters and intitial conditions
y = c(5,5)
p = c(0.5,0.02,0.1,0.2)
times=seq(from=1,to=100,by=0.1)

# Simulate the model using ode
modelSim=ode(y=y,times=times,func=predPreySim,parms=p)
colnames(modelSim)[2] <- "herbivore"
colnames(modelSim)[3] <- "predator"

# save model as a data frame for plotting purposes
modelSim2 <- as.data.frame(modelSim)

# plot simulation
ggplot(data=modelSim2) + geom_line(aes(x=time,y=herbivore),color="firebrick")+geom_line(aes(x=time,y=pre
```




```
#show result summary of the data frame
kable(summary(modelSim))
```

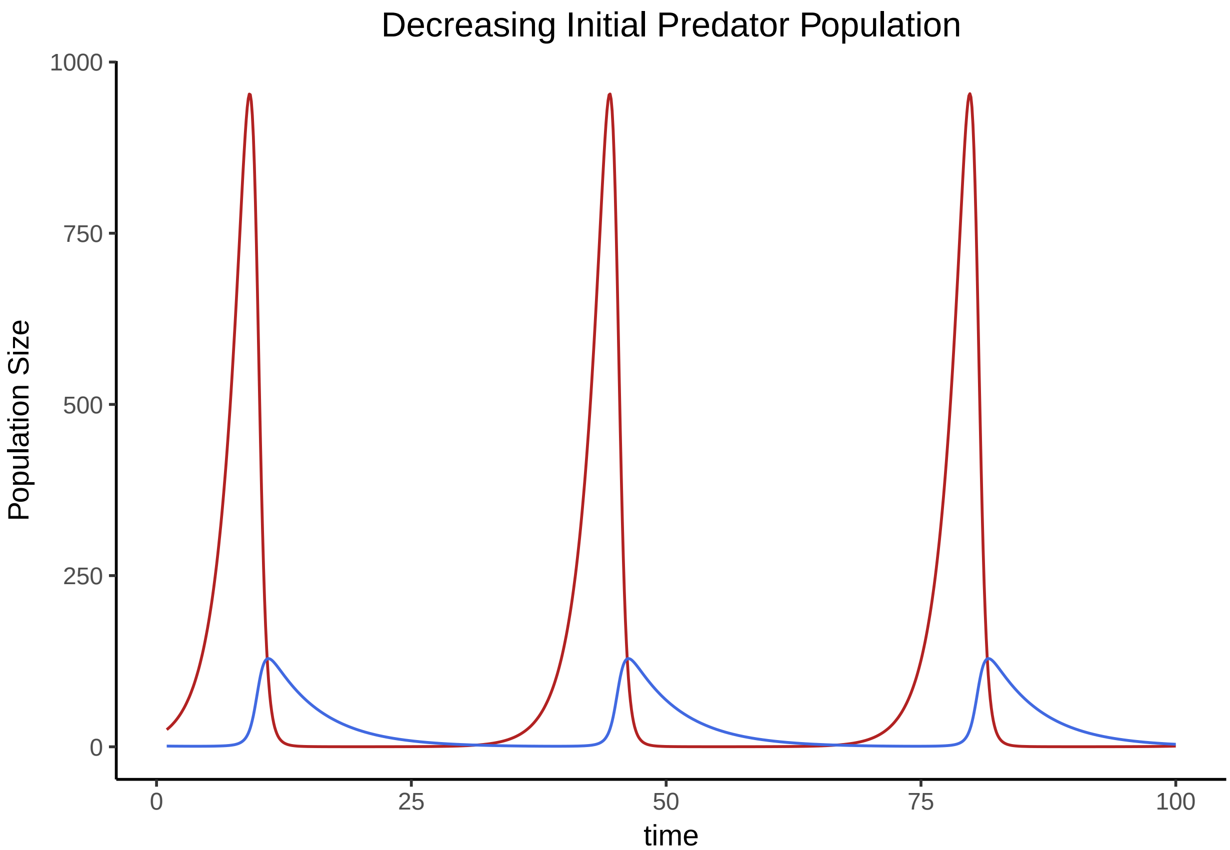
	herbivore	predator
Min.	0.6326619	1.952785
1st Qu.	1.4337351	2.905456
Median	10.1132523	8.216084
Mean	97.7414364	22.871279
3rd Qu.	94.6878219	32.804991
Max.	701.7623894	100.467088
N	991.0000000	991.000000
sd	177.0075387	28.422647

The above two models changed the number of prey from 25 to 125 and from 25 to 5, increasing and decreasing the number of prey, respectively. Changing the prey number does not change much of the overall model. There are some slight changes in overall min/max of predator/prey, however the frequency does not seem to change at all. Interestingly, decreasing the inital number of prey increased the maximum number of prey and predators. The relative numbers between the predator and prey does not change much.

Changing the number of predators

Now let's do the same thing for the predators as we did for the prey above. Change predator initial values from 5 to 1.

```r
# simulate dynamics by resetting parameters and intitial conditions
y = c(25,1)
p = c(0.5,0.02,0.1,0.2)
times=seq(from=1,to=100,by=0.1)

# simulate the model using ode
modelSim=ode(y=y,times=times,func=predPreySim,parms=p)
colnames(modelSim)[2] <- "herbivore"
colnames(modelSim)[3] <- "predator"

# save model as a data frame for plotting purposes
modelSim2 <- as.data.frame(modelSim)

# plot simulation
ggplot(data=modelSim2) + geom_line(aes(x=time,y=herbivore),color="firebrick")+geom_line(aes(x=time,y=pr
```



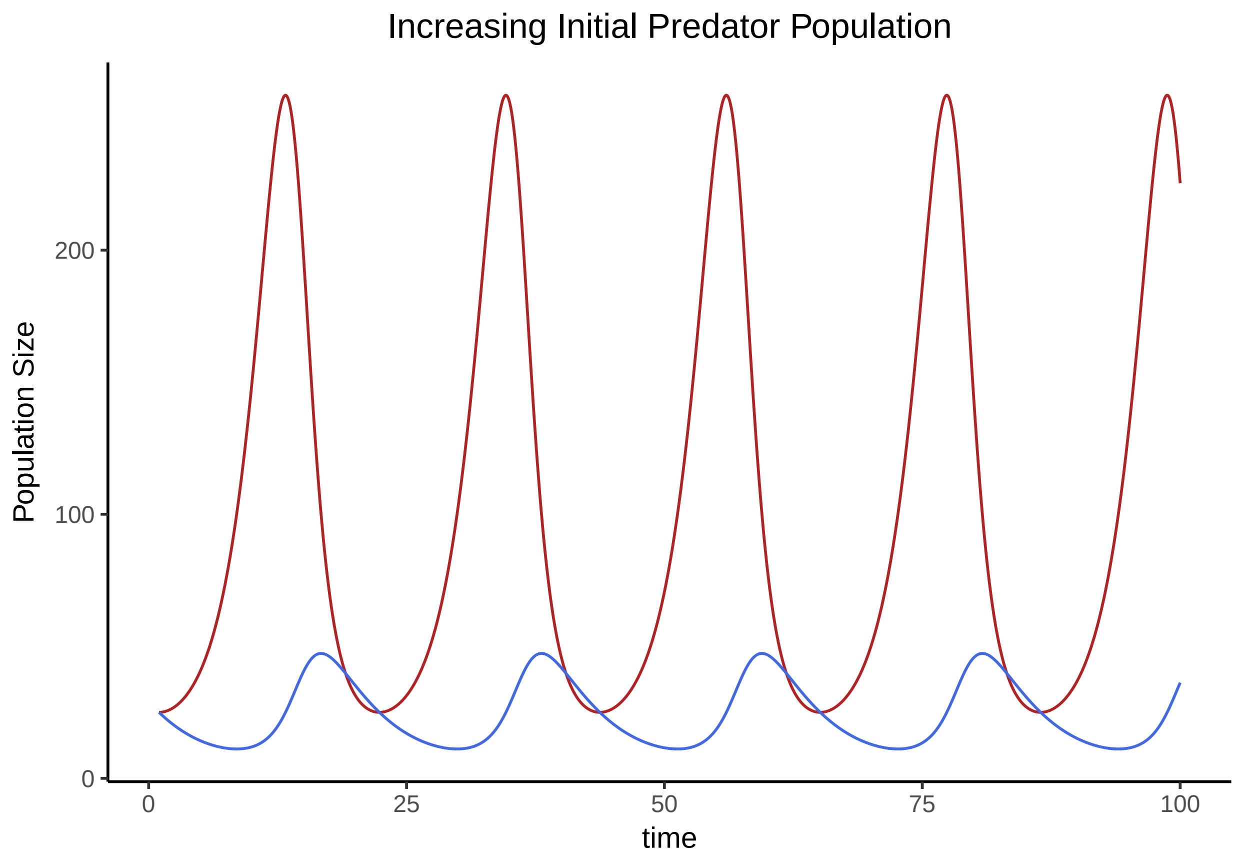

```r
#show result summary of the data frame
kable(summary(modelSim))
```

	herbivore	predator
Min.	0.0687258	0.7681323
1st Qu.	0.1566069	1.7932943
Median	1.5085196	8.6631537
Mean	106.4375884	26.5960674
3rd Qu.	63.4302043	37.5351994
Max.	953.8490768	128.8241713
N	991.0000000	991.0000000

	herbivore	predator
sd	227.6725557	35.6732609

Change predator initial values from 5 to 25.

```
# simulate dynamics by resetting parameters and intitial conditions
y = c(25,25)
p = c(0.5,0.02,0.1,0.2)
times=seq(from=1,to=100,by=0.1)

# Simulate the model using ode
modelSim=ode(y=y,times=times,func=predPreySim,parms=p)
colnames(modelSim)[2] <- "herbivore"
colnames(modelSim)[3] <- "predator"

# save model as a data frame for plotting purposes
modelSim2 <- as.data.frame(modelSim)

# plot simulation
ggplot(data=modelSim2) + geom_line(aes(x=time,y=herbivore),color="firebrick")+geom_line(aes(x=time,y=pre
```




```
#show result summary of the data frame
kable(summary(modelSim))
```

	herbivore	predator
Min.	25.00000	11.12827

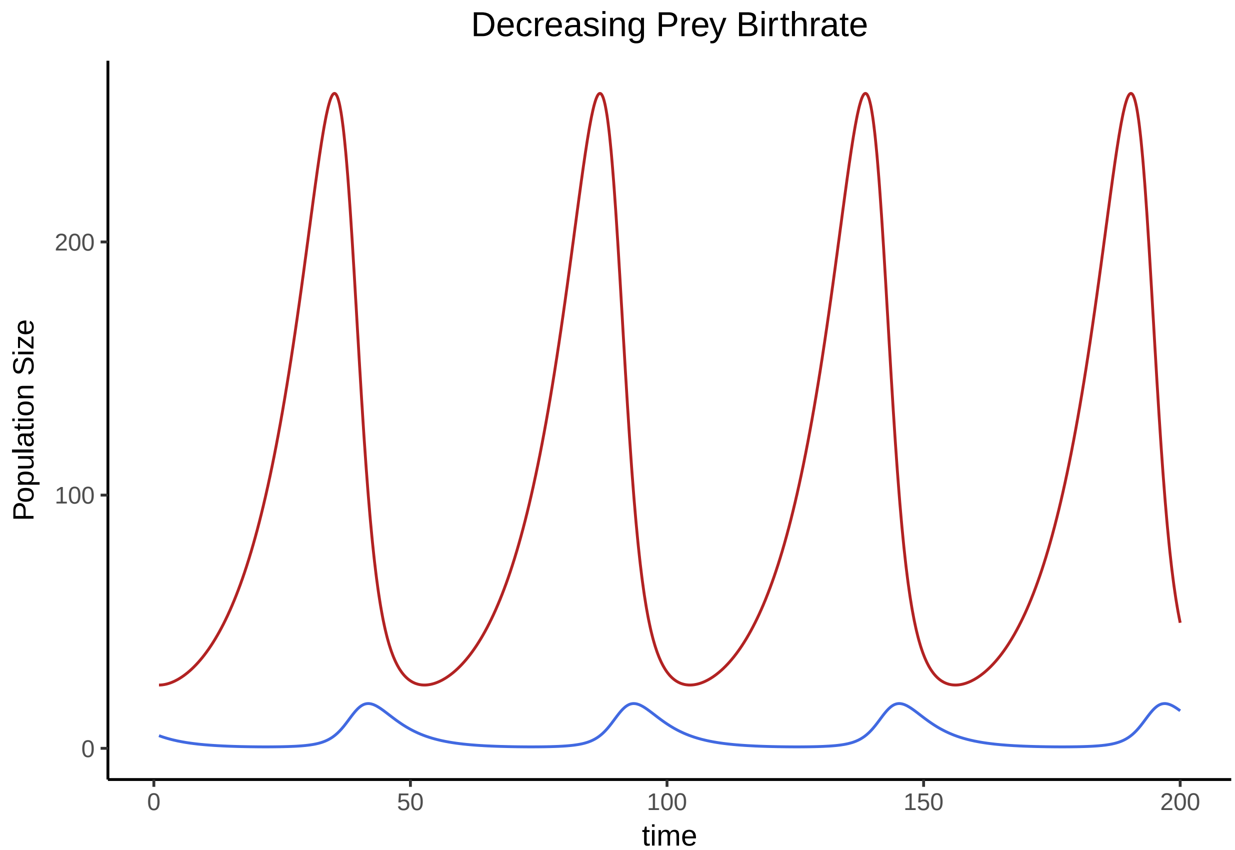
	herbivore	predator
1st Qu.	32.82907	13.07874
Median	67.63201	19.72100
Mean	101.89641	23.89643
3rd Qu.	165.16061	33.56128
Max.	258.66607	47.30765
N	991.00000	991.00000
sd	79.45680	12.04776

Unlike changing the intial number of prey, changing the inital number of predators dramatically changes the model ouptut. For starters, the frequency for both prey/predator are decreased/increased when the inital predator count is decreased/increased. Additionally, the inital number of predators seems to have a bigger impact on both max/min numbers of both predator/prey. The relative numbers between the predator/prey does not change much.

Changing the prey birthrate

Change prey birthrate from 0.5 to 0.1 (and time to 200)

```
# simulate dynamics by resetting parameters and intitial conditions
y = c(25,5)
p = c(0.1,0.02,0.1,0.2)
times=seq(from=1,to=200,by=0.1)

# Simulate the model using ode
modelSim=ode(y=y,times=times,func=predPreySim,parms=p)
colnames(modelSim)[2] <- "herbivore"
colnames(modelSim)[3] <- "predator"

# save model as a data frame for plotting purposes
modelSim2 <- as.data.frame(modelSim)

# plot simulation
ggplot(data=modelSim2) + geom_line(aes(x=time,y=herbivore),color="firebrick")+geom_line(aes(x=time,y=pre
```



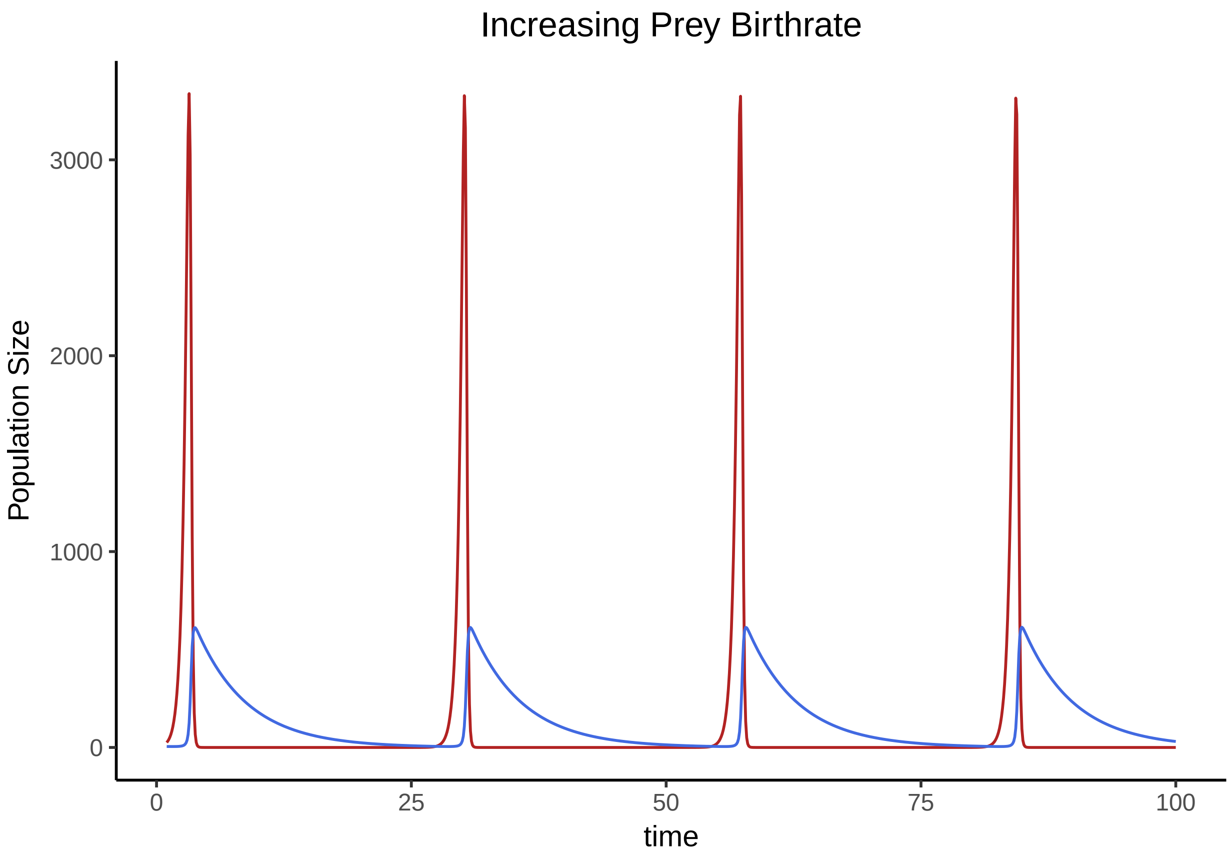

```r
#show result summary of the data frame
kable(summary(modelSim))
```

	herbivore	predator
Min.	24.99999	0.5784130
1st Qu.	34.98699	0.8090509
Median	72.61223	2.0053024
Mean	102.69778	4.8304693
3rd Qu.	161.48185	7.2861356
Max.	258.66634	17.6770091
N	1991.00000	1991.0000000
sd	77.14174	5.4445853

Change prey birthrate from 0.5 to 2.5

```r
# simulate dynamics by resetting parameters and intitial conditions
y = c(25,5)
p = c(2.5,0.02,0.1,0.2)
times=seq(from=1,to=100,by=0.1)

# Simulate the model using ode
modelSim=ode(y=y,times=times,func=predPreySim,parms=p)
colnames(modelSim)[2] <- "herbivore"
colnames(modelSim)[3] <- "predator"

# save model as a data frame for plotting purposes
```

```
modelSim2 <- as.data.frame(modelSim)

# plot simulation
ggplot(data=modelSim2) + geom_line(aes(x=time,y=herbivore),color="firebrick")+geom_line(aes(x=time,y=pre
```




```
#show result summary of the data frame
kable(summary(modelSim))
```

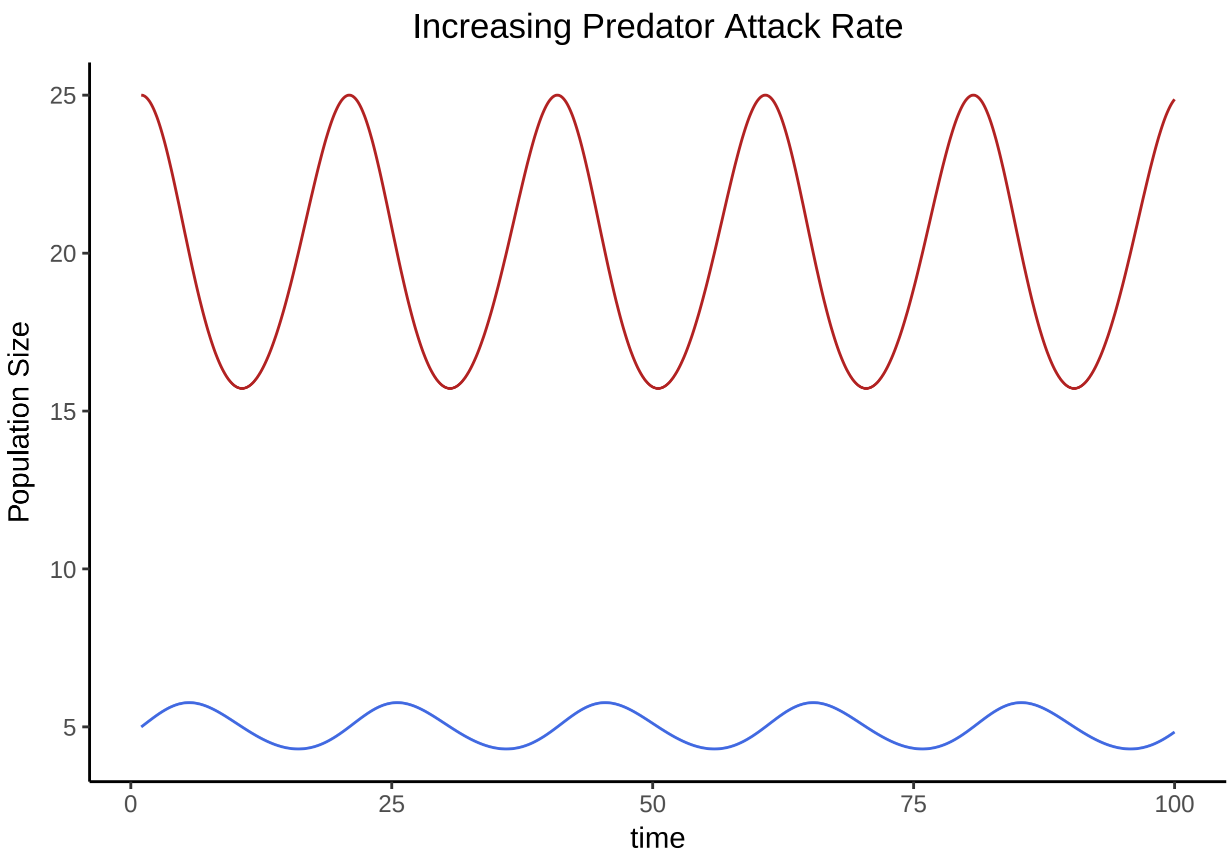
	herbivore	predator
Min.	0.0000000	4.693959
1st Qu.	0.0000000	15.682254
Median	0.0000001	60.056245
Mean	109.0432336	135.143648
3rd Qu.	0.0125971	199.813250
Max.	3338.0019282	613.872554
N	991.0000000	991.000000
sd	467.6765701	161.949298

Changing prey birthrate has significant effects on both max/min of predator prey and also of the frequency of both. Decreasing prey birthrate significantly decreased both frequency and max/min of both predator/prey, much more than changing the inital numbers of prey/predator. Additionally, the prey birthrate seems to havve a strong effect on the relative numbers between prey/predator, as shown by the differences in mean between the two. Additionally, increasing prey birthrate causes an interesting phenomenon where most of the time the prey are mostly dead but shoot up exponentially every now and then, whilst decreasing birthrate causes a more conservative change in the number of prey, more like a sinusoidal curve.

Changing predator attack rate

Change predator attack rate from 0.02 to 0.1

```r
# simulate dynamics by resetting parameters and intitial conditions
y = c(25,5)
p = c(0.5,0.1,0.1,0.2)
times=seq(from=1,to=100,by=0.1)

# Simulate the model using ode
modelSim=ode(y=y,times=times,func=predPreySim,parms=p)
colnames(modelSim)[2] <- "herbivore"
colnames(modelSim)[3] <- "predator"

# save model as a data frame for plotting purposes
modelSim2 <- as.data.frame(modelSim)

# plot simulation
ggplot(data=modelSim2) + geom_line(aes(x=time,y=herbivore),color="firebrick")+geom_line(aes(x=time,y=pre
```



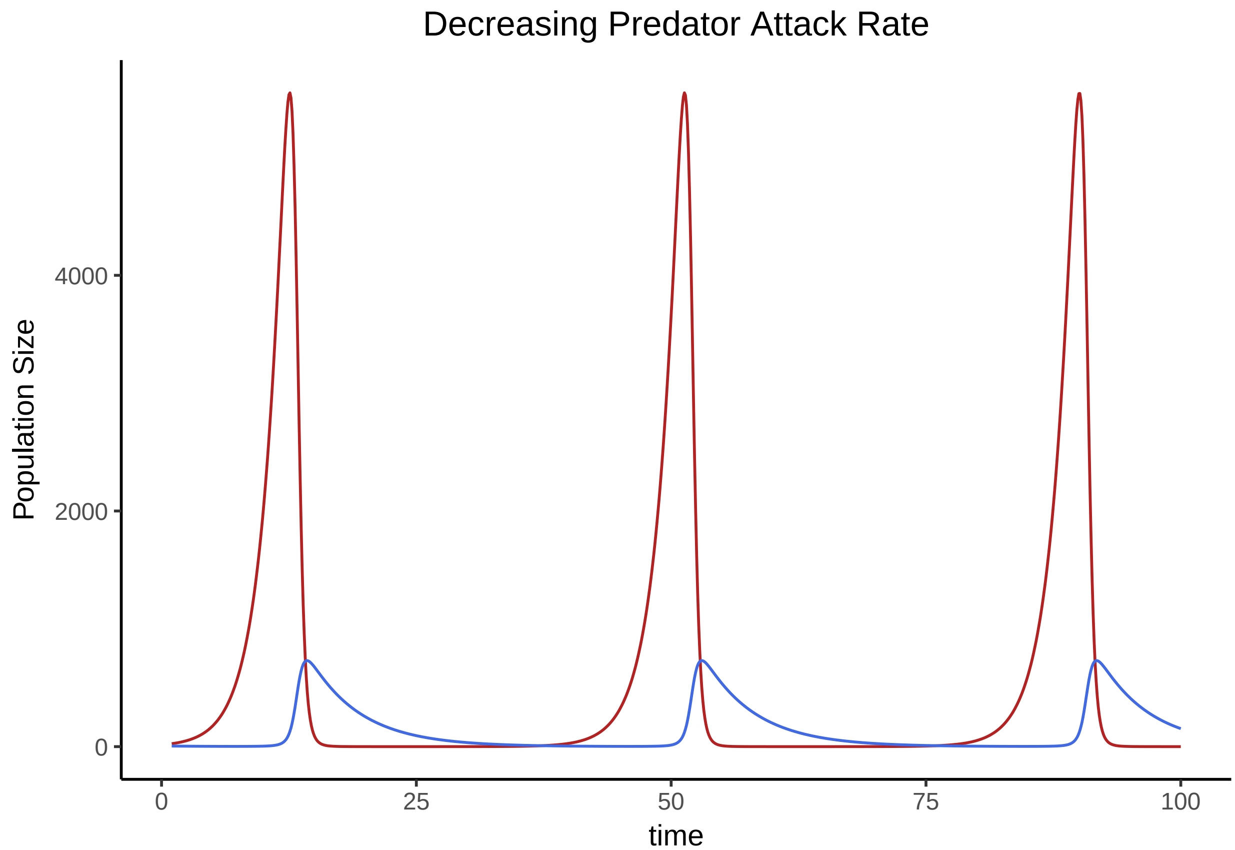

```r
#show result summary of the data frame
kable(summary(modelSim))
```

	herbivore	predator
Min.	15.715773	4.3024738
1st Qu.	16.744698	4.4830608
Median	19.618941	4.9686235
Mean	19.972108	5.0004559

	herbivore	predator
3rd Qu.	23.192277	5.5175409
Max.	25.000000	5.7691227
N	991.000000	991.0000000
sd	3.264623	0.5194991

Change predator attack rate from 0.02 to 0.004

```
# simulate dynamics by resetting parameters and intitial conditions
y = c(25,5)
p = c(0.5,0.004,0.1,0.2)
times=seq(from=1,to=100,by=0.1)

# Simulate the model using ode
modelSim=ode(y=y,times=times,func=predPreySim,parms=p)
colnames(modelSim)[2] <- "herbivore"
colnames(modelSim)[3] <- "predator"

# save model as a data frame for plotting purposes
modelSim2 <- as.data.frame(modelSim)

# plot simulation
ggplot(data=modelSim2) + geom_line(aes(x=time,y=herbivore),color="firebrick")+geom_line(aes(x=time,y=pre
```




```
#show result summary of the data frame
kable(summary(modelSim))
```

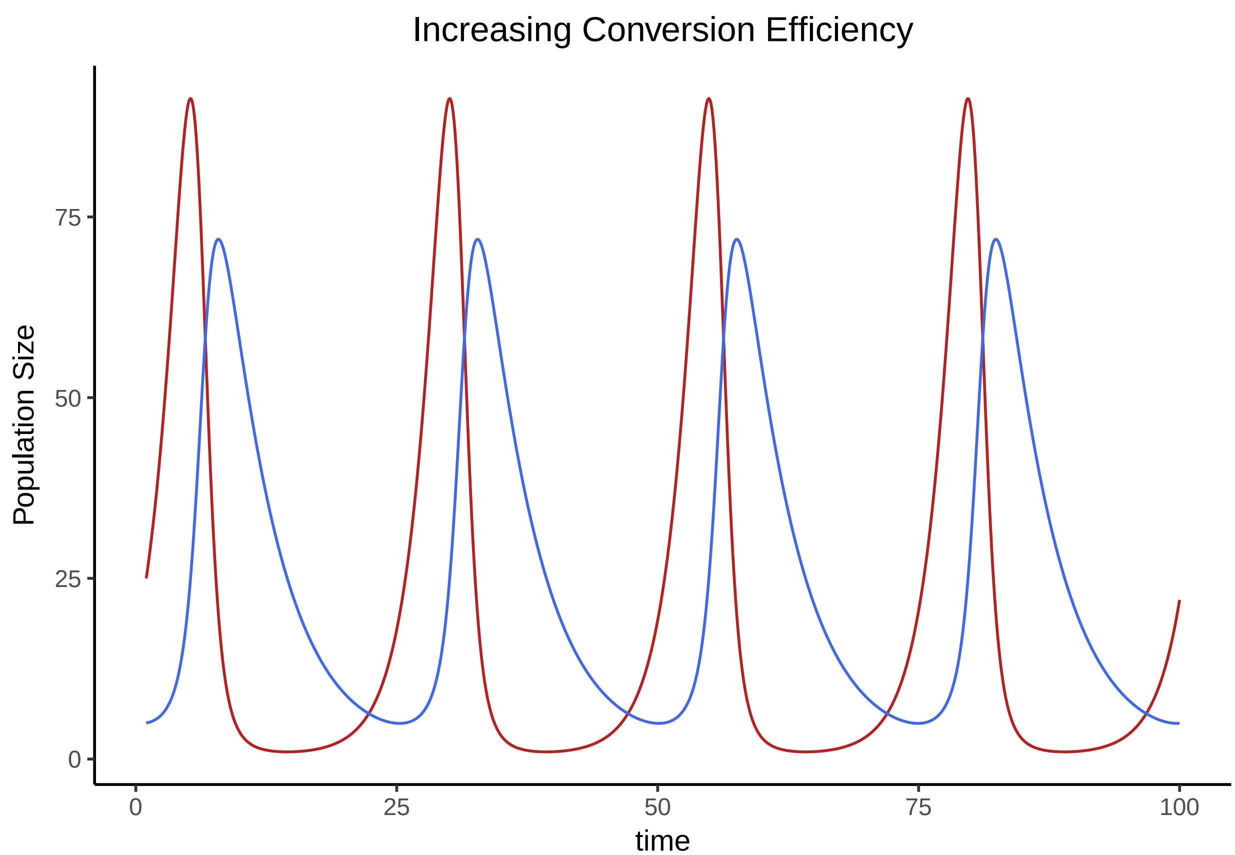
	herbivore	predator
Min.	0.0839023	2.156335
1st Qu.	0.3204918	4.319488
Median	9.4939063	27.365309
Mean	585.7889308	139.188351
3rd Qu.	313.3331253	203.998745
Max.	5549.1553765	730.282173
N	991.0000000	991.000000
sd	1287.3686222	202.618168

Changing predator attack rate has significant effects on the overall model. Both the max/min and frequecy of predator/prey are significantly affected. Increasing the attack rate significantly decreases both the number of predators and prey and also increases the frequency of both predator/prey cycle, whilst decreasing the attack rate does the opposite. Additonally, decreasing the attack rate brings the prey to near extinction levels, as evidenced by the 0.08 minimum of prey, perhaps making it an unsustainable model. However, the relationship between the means does not change much from the original model (~1:4 ratio for predator:prey).

Changing conversion efficiency

Change conversion efficiency from 0.1 to 0.5

```
# simulate dynamics by resetting parameters and intitial conditions
y = c(25,5)
p = c(0.5,0.02,0.5,0.2)
times=seq(from=1,to=100,by=0.1)

# Simulate the model using ode
modelSim=ode(y=y,times=times,func=predPreySim,parms=p)
colnames(modelSim)[2] <- "herbivore"
colnames(modelSim)[3] <- "predator"

# save model as a data frame for plotting purposes
modelSim2 <- as.data.frame(modelSim)

# plot simulation
ggplot(data=modelSim2) + geom_line(aes(x=time,y=herbivore),color="firebrick")+geom_line(aes(x=time,y=pre
```



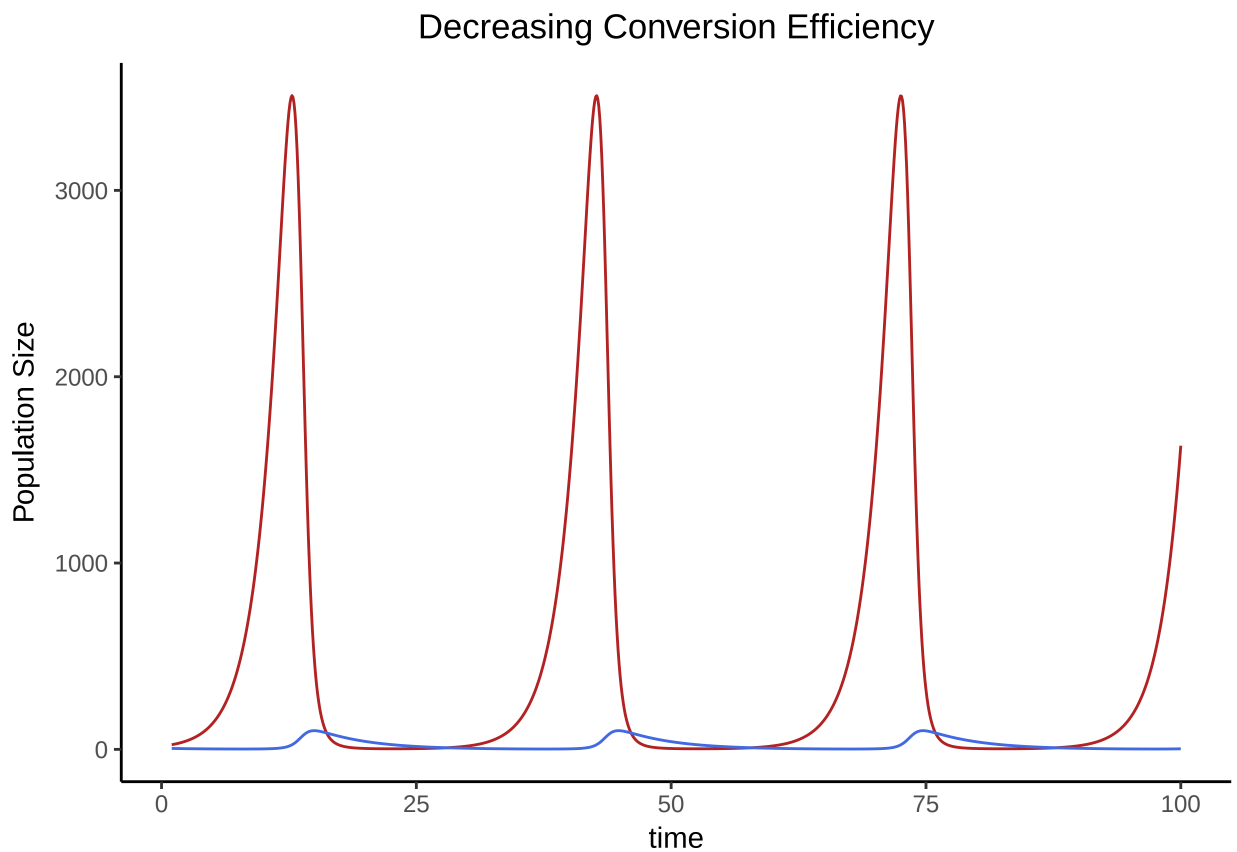

```
#show result summary of the data frame
kable(summary(modelSim))
```

	herbivore	predator
Min.	0.9950496	4.933426
1st Qu.	1.5542510	7.059209
Median	5.2672934	15.703786
Mean	19.9923981	25.043921
3rd Qu.	28.4812166	39.363239
Max.	91.3990109	71.922212
N	991.0000000	991.000000
sd	27.1966963	21.480734

Change conversion efficiency from 0.1 to 0.02

```
# simulate dynamics by resetting parameters and intitial conditions
y = c(25,5)
p = c(0.5,0.02,0.02,0.2)
times=seq(from=1,to=100,by=0.1)

# Simulate the model using ode
modelSim=ode(y=y,times=times,func=predPreySim,parms=p)
colnames(modelSim)[2] <- "herbivore"
colnames(modelSim)[3] <- "predator"

# save model as a data frame for plotting purposes
```

```r
modelSim2 <- as.data.frame(modelSim)

# plot simulation
ggplot(data=modelSim2) + geom_line(aes(x=time,y=herbivore),color="firebrick")+geom_line(aes(x=time,y=pre
```




```r
#show result summary of the data frame
kable(summary(modelSim))
```

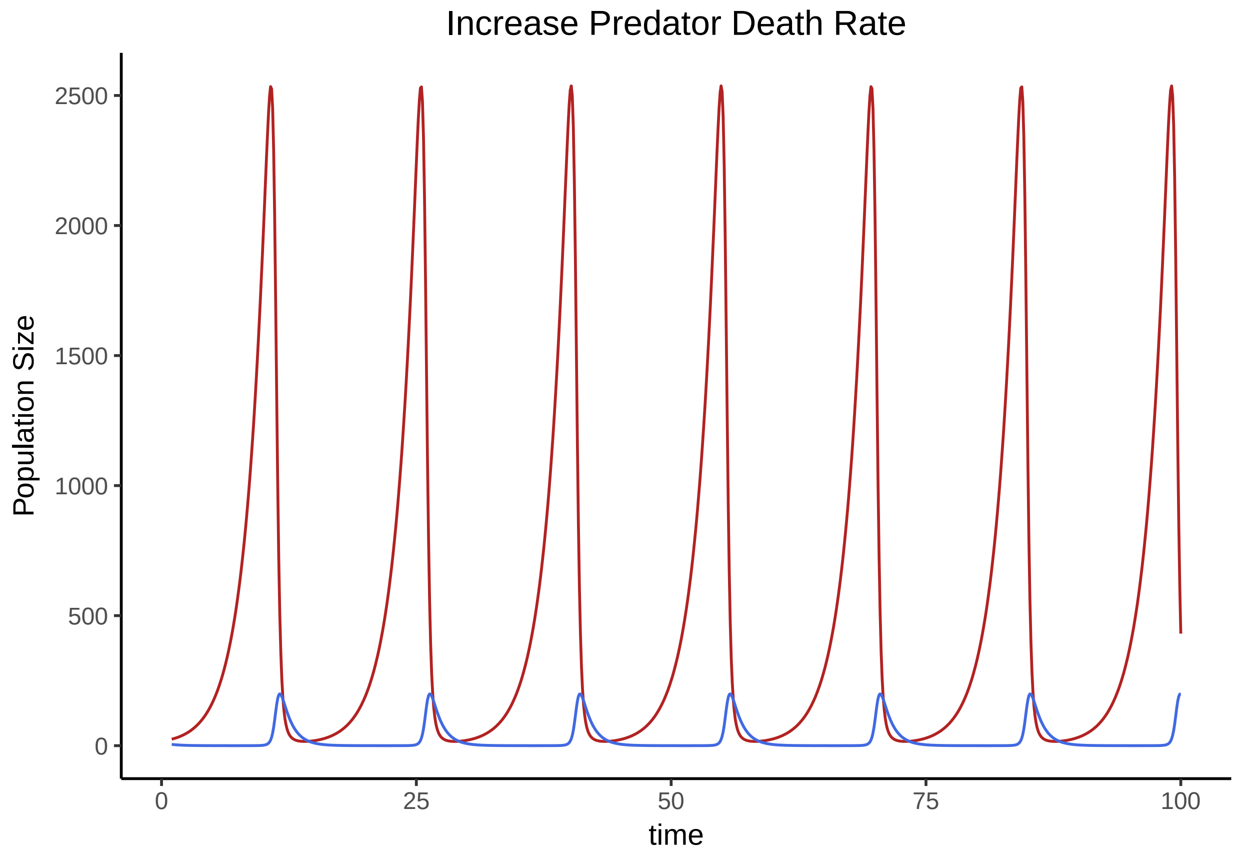
	herbivore	predator
Min.	3.163315	1.952785
1st Qu.	7.168682	2.905456
Median	50.566255	8.216081
Mean	488.707225	22.871278
3rd Qu.	473.439119	32.804981
Max.	3508.812000	100.467086
N	991.000000	991.000000
sd	885.037682	28.422646

Changing conversion efficiency also changes min/max, frequency, and predator/prey relationships. However, the most interesting aspect of changing the conversion efficency is the change in relative means between the prey/predator relationship. Increasing conversion efficiency brings the relative number of prey much closer to the relative mean of the predator, as would be expected because the predator can more efficiently convert the prey into more predators.

Changing predator death rate

Change predator death rate from 0.2 to 1

```
# simulate dynamics by resetting parameters and intitial conditions
y = c(25,5)
p = c(0.5,0.02,0.1,1)
times=seq(from=1,to=100,by=0.1)

# Simulate the model using ode
modelSim=ode(y=y,times=times,func=predPreySim,parms=p)
colnames(modelSim)[2] <- "herbivore"
colnames(modelSim)[3] <- "predator"

# save model as a data frame for plotting purposes
modelSim2 <- as.data.frame(modelSim)

# plot simulation
ggplot(data=modelSim2) + geom_line(aes(x=time,y=herbivore),color="firebrick")+geom_line(aes(x=time,y=pre
```



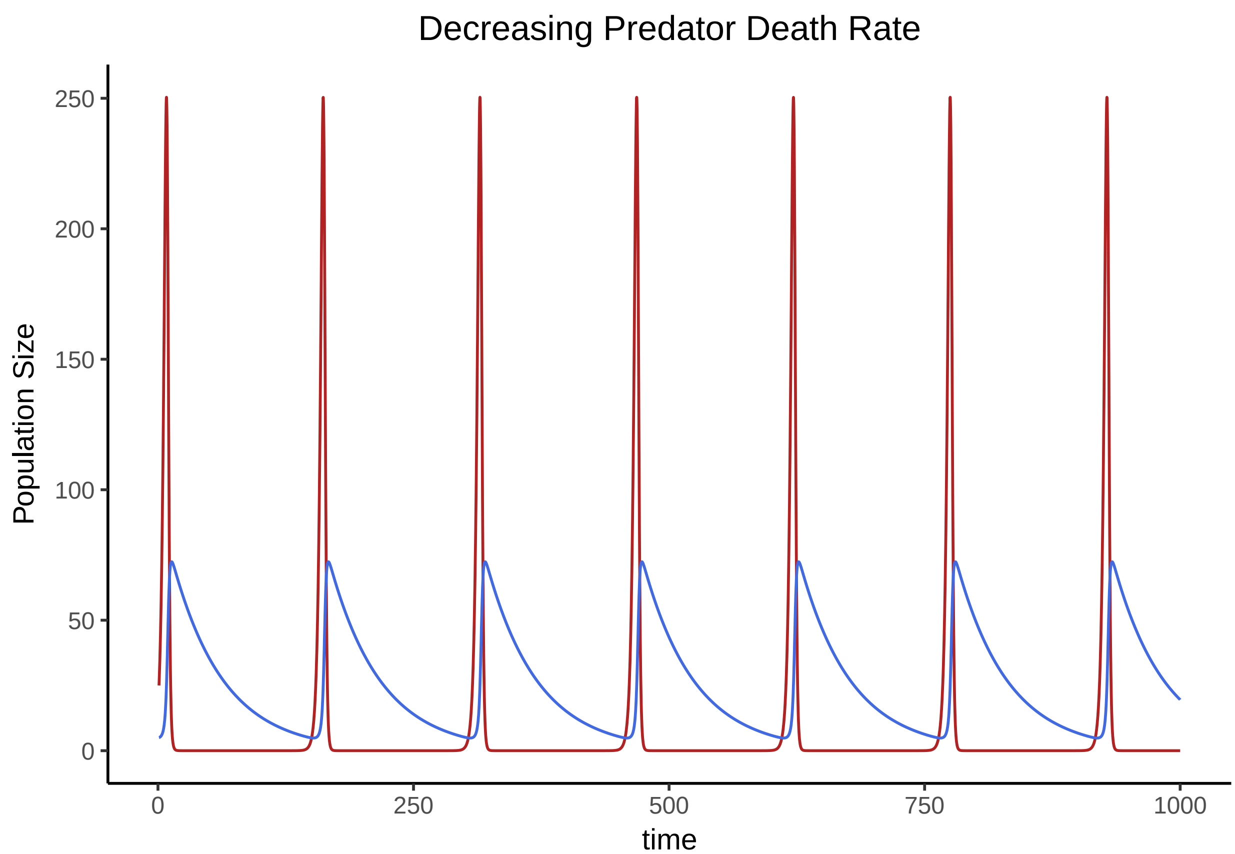

```
#show result summary of the data frame
kable(summary(modelSim))
```

	herbivore	predator
Min.	16.39540	0.0686149
1st Qu.	36.67678	0.1454281
Median	154.09082	1.0066230
Mean	518.29438	23.6410432

	herbivore	predator
3rd Qu.	722.83426	15.0825614
Max.	2536.99386	199.4270987
N	991.00000	991.0000000
sd	707.82100	48.7469976

Change predator death rate from 0.2 to 0.04

```
# simulate dynamics by resetting parameters and intitial conditions
y = c(25,5)
p = c(0.5,0.02,0.1,0.02)
times=seq(from=1,to=1000,by=0.1)

# Simulate the model using ode
modelSim=ode(y=y,times=times,func=predPreySim,parms=p)
colnames(modelSim)[2] <- "herbivore"
colnames(modelSim)[3] <- "predator"

# save model as a data frame for plotting purposes
modelSim2 <- as.data.frame(modelSim)

# plot simulation
ggplot(data=modelSim2) + geom_line(aes(x=time,y=herbivore),color="firebrick")+geom_line(aes(x=time,y=pre
```




```
#show result summary of the data frame
kable(summary(modelSim))
```

	herbivore	predator
Min.	0.0000000	4.856713
1st Qu.	0.0000000	9.078601
Median	0.0000034	19.836287
Mean	10.6819878	26.101291
3rd Qu.	0.0170270	39.313460
Max.	250.3980394	72.404377
N	9991.0000000	9991.000000
sd	40.4306292	19.789751

Predator death rate also has significant effects on the max/min and frequency in predator/prey. Additionally, the relative means between the predator/prey relationships is also affected. Increasing predator death rate increases prey counts (both max and mean) but also increases the max predator count. Decreasing predator death rate has the opposite effect while both increase the frequency.

Conclusion

Overall, all of the state variables and parameters have some effect on the model, whether they change the max/min of the prey/predator, change the overall frequency, change the rate of increase/decrease, or change the relative means of the two populations. What is interesting to note are the models in which the predator/prey minimun numbers reach <1. Realistically, it seems impossible to have half a predator or half a prey, suggesting these models are unsustainable. This suggests, that especially when changing the parameters, one must be careful that the parameters are not changed significantly since the model may become unrealistic, unless that is the purpose.

Questions

What can you say about the "role" of each parameter? What is the relationship between parameter values and predator-prey cycle length?

Each parameter affects the model in different ways, with some parameters having larger effects than others depending on which parameters you are interested in. Dramatically changing some of the parameters makes the model almost unsustainable, as evidenced by the minimum numbers of prey/predator in each summary statement that are less than 1. Otherwise, the role of each parameter is fairly straightforward. For starters, what needs to be understood is that predator/prey relationships are intertwined, as evidenced by their rate equations being dependent on the population size of the other.

The prey birthrate has a significant effect on the number of prey in the model, which inherently makes sense. Since the prey birthrate is the only factor that increases the number of prey (assuming the second factor will always be negative or zero), increasing/decreasing this number understandably increase/decreases the number of prey. However, increasing the prey birthrate by a factor of five does not increase the number of prey by a factor of five, because the change in the number of prey (dp/dt), is limited by the number of prey also. Since the number of predators is reliant on the number of prey available, increasing the prey birth rate also increases the predator count. However, increasing the prey birthrate causes an exponential increase and sudden crash in population numbers caused by the exponential increase in predator numbers, suggesting that increasing the prey birthrate may be unsustainable. Decreasing the prey birthrate has the opposite effect, however, the model seems more sustainable. Additionally, we do not see the exponential increase and crash of the prey but instead see a more steady increase and decrease in prey numbers and a similar trend with the predator numbers. Prey birthrate seems to significantly affect the frequency of the prey/predator model, where decreasing prey birthrate decreases the frequency while increasing prey birth rate increases the frequency.

Changing the predator attack rate has a significant effect on both prey and predator numbers in the model, as expected. Increasing attack rate significantly decreases the number of prey, and as a result, the maximum number of predators decreases. However, the mean predator count is very close to the original model, most likely due to the dependence of the predator numbers on the prey numbers. Increased predator attact rate increases the frequency of prey/predator turnaround and also creates a steady, sinusoidal model. Decreasing attack rate has the opposite effect with exponential increase/decrease of predator/prey population size while also lowering the frequency.

The conversion efficiency is interesting, since it is one of the few parameters that significantly affects the ratios between predator and prey population sizes. Increasing conversion efficiency significantly increases the number of predators relative to the number of prey while still limiting the population size of the prey, and vice versa. The conversion efficiency does not seem to have a significant effect on the frequency of the predator/prey cycle.

Finally, the predator death rate also has effects as expected. Increasing predator death rate causes a surge in prey numbers until a point where we see a significant enough population of predators to bring the prey population down. Increasing predator death rate also increases the predator/prey cycle frequency. Decreasing the predator death rate has the opposite effect, where we observe a sudden spike in prey numbers when the predator numbers decrease. Due to the predator lingering (because of their lower death rate), prey essentially go extinct until there are low enough numbers of predators that the prey can recover. The lingering of the predators causes a decrease in frequency of the predator/prey cycle, as expected.

What can you say about the role of predators in the simulations?

The role of predators in the simulations is to keep in check the population of prey when the number of prey get too high. Without a limiting constraint on the prey, such as the predator, the prey would hypothetically expand exponentially unless there were to be a limiting resource or outside competition. This fact can be observed from the fact that in any of the given models above, the predator numbers only rise when the prey number rises, which inherently makes sense if we observe the equations. Because in the Lotka-Volterra model the only factor that allows predator numbers to increase is $e * a * P * H$ (assuming $s * P$ is always negative), the number of prey in the model (H) is the limiting factor in the model. And only when the number of predators rises the prey numbers start to decrease, and thus are kept in check. Once the number of prey die down, $e * a * P * H$ goes down and $s * P$ takes over, causing the overall number of predators to decrease.

Group Project Part 2 - Rosenzweig-MacArthur Model

Model Description

The Rosenzweig-MacArthur model describes predator-prey dynamics. It is based off of the simpler Lotka-Volterra Model, but it adds two elements:

Prey become self-limiting in absence of predators.

Predator kill density cannot grow indefinitely; there reaches a maximum no matter how large the prey density is.

Model Equations

Rosenzweig and MacArthur described the prey population (H) and the predator population (P) with the following equations:

$\frac{dh}{dt} = bH(1 - \alpha H) - \frac{wHP}{(d+H)}$

$\frac{dP}{dt} = \frac{ewHP}{(d+H)} - sP$

Qualitative Results

There are three qualitative results for this model:

1. Prey go to carrying capacity, predators go extinct.

$H^* = \frac{1}{\alpha} = k$ and $P^* = 0$

2. Prey and predators coexist in stable equilibrium.

$H^* = \frac{sd}{(ew-s)}$ and $P^* = \frac{bH^* - bd\alpha - b\alpha(H^*)^2 + db}{w}$

3. Prey and predators coexist in an oscillatory manner.

Let's begin by writing a function that describes the model:

```
### Function for Rosenzweig-MacArthur Predator/Prey Equations ###
ddRM <- function (t,y,p){

  #Unpack State Variables
  H = y[1] #Prey
  P = y[2] #Predator

  #Unpack Model Parameters
  b = p[1] #Prey birth rate
  e = p[2] #Conversion efficency of prey to predator
  s = p[3] #Predator death rate
  w = p[4] #Predator attack rate
  d = p[5] #Density of prey when predator's kill rate is 1/2*max
  alpha = p[6] #Prey self-limiting term

  #Defining Rates of Change
  dHdt = b*H*(1 - alpha*H) - w*H/(d + H)*P
  dPdt = e*w*H/(d + H)*P- s*P

  return(list(c(dHdt, dPdt)))
}
```

Model Solution

Now, we will simulate the model with the following initial parameters:

$b = 0.8$

$e = 0.07$

$s = 0.2$

$w = 5$

$d = 400$

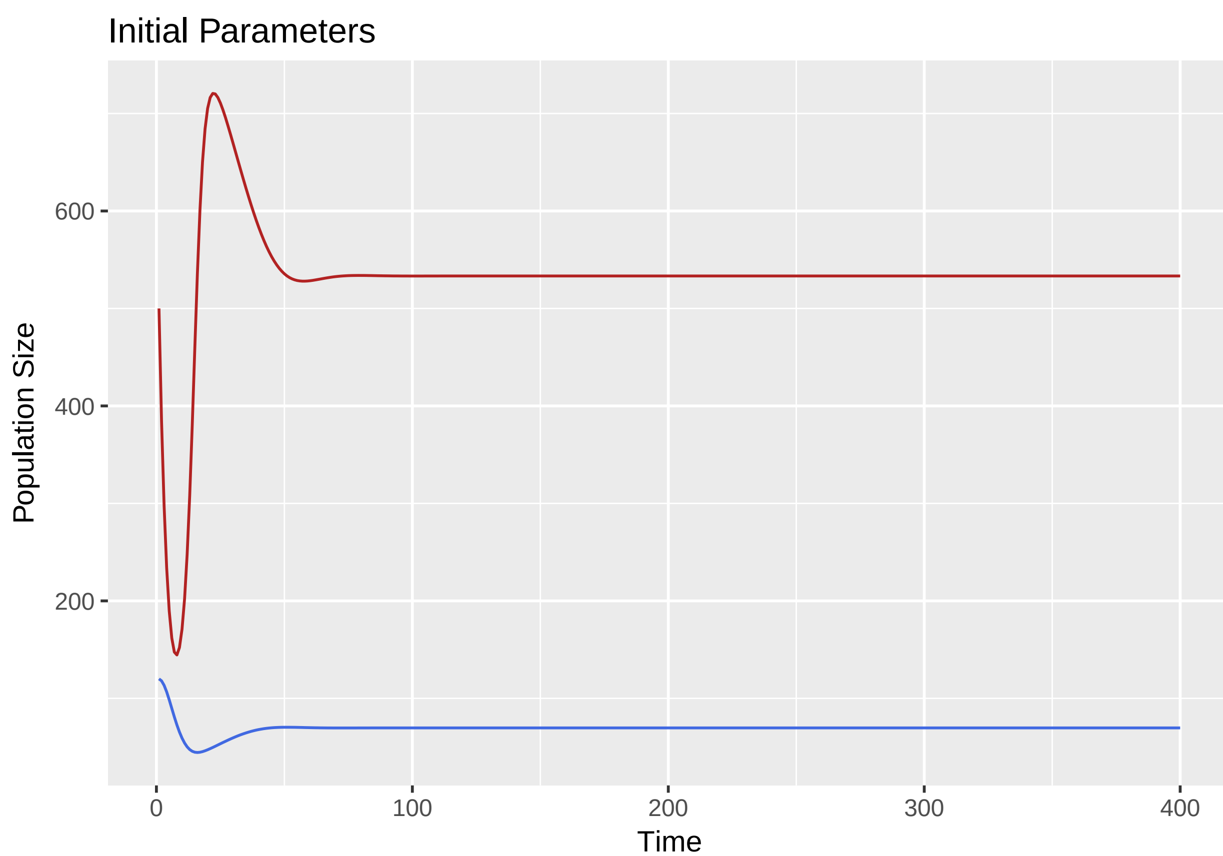
$\alpha = 0.001$

$H0 = 500$

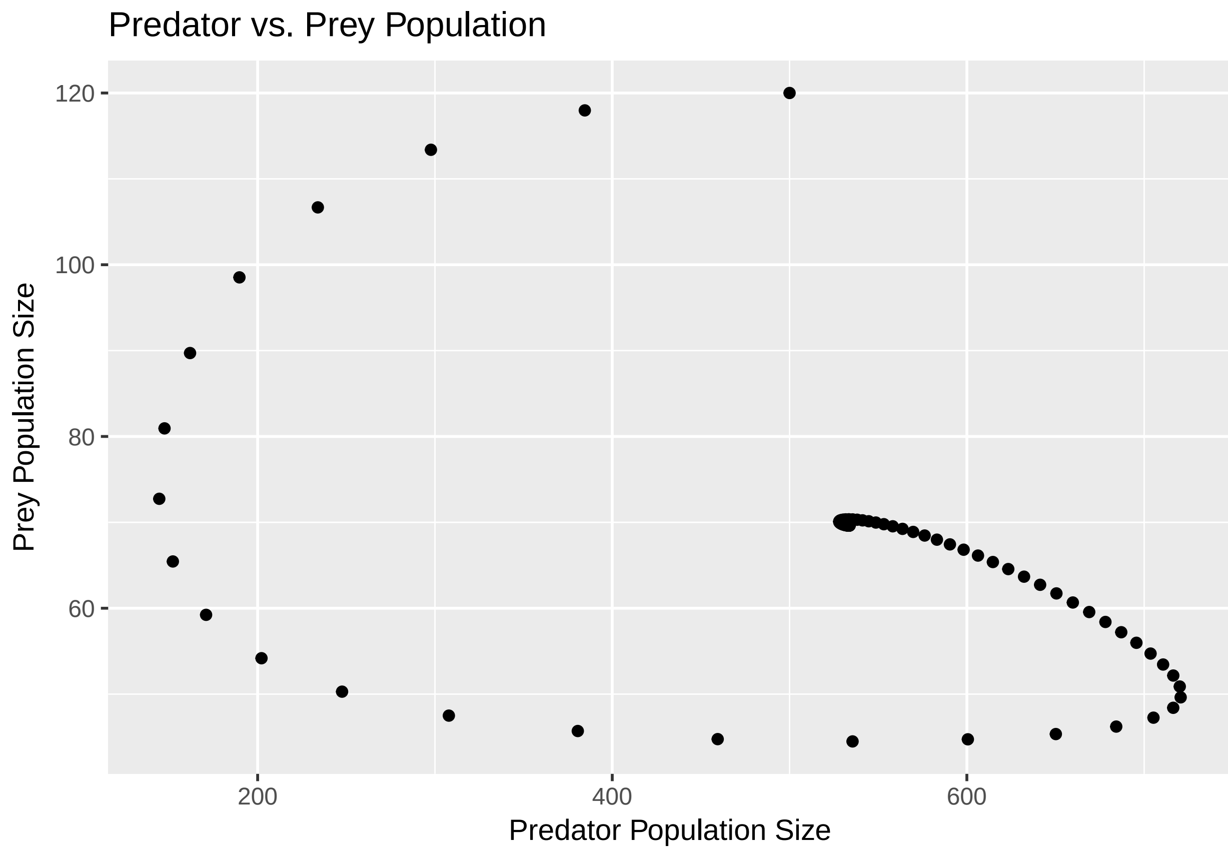
$P0 = 120$

```
### Defining Initial Parameters ###
b = 0.8
e = 0.07
```

```r
s = 0.2
w = 5
d = 400
alpha = 0.001
params = c(b, e, s, w, d, alpha)

H0 = 500
P0 = 120
Y0 = c(H0, P0)

times = 1:400

modelSim = ode(y=Y0, times = times, func = ddRM, parms = params)

#Subsetting the Results
modelOutput = data.frame(time=modelSim[,1],H=modelSim[,2],P = modelSim[,3])

#Graphing the Results
ggplot(modelOutput) +
  geom_line(aes(x = time, y = H), colour = "firebrick") +
  geom_line(aes(x = time, y = P), colour = "royalblue") +
  ggtitle("Initial Parameters") +
  labs(y="Population Size", x = "Time")
```




```r
ggplot(modelOutput, aes(H,P)) + geom_point() +
  ggtitle("Predator vs. Prey Population") +
  labs(y="Prey Population Size", x = "Predator Population Size")
```



Describing Role of Parameters

Let's now change individual parameters in order to show the role that each one has

Predator Attack Rate

Low Attack Rate:

Low attack rate means that predators cannot sustain themselves, and their population goes to zero

Prey population reach self-limiting equilibrium

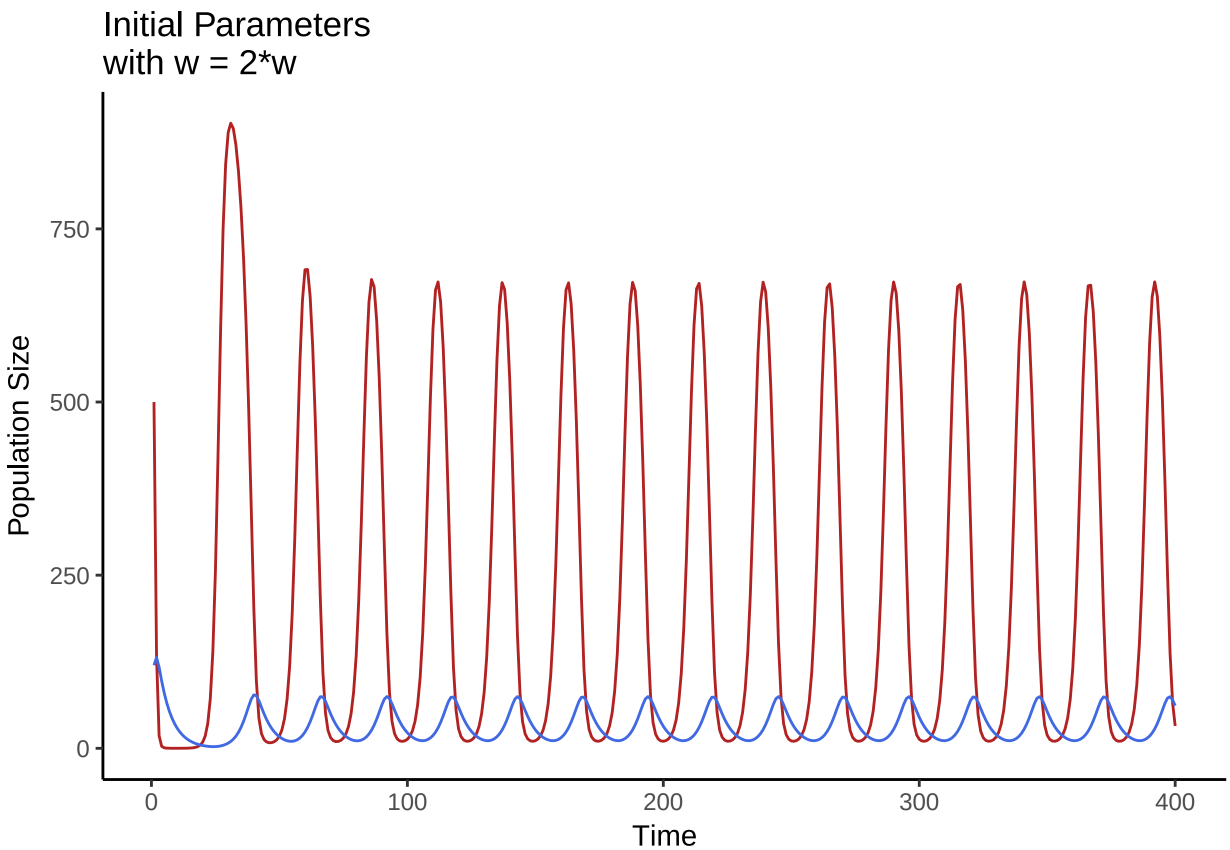
High Attack Rate

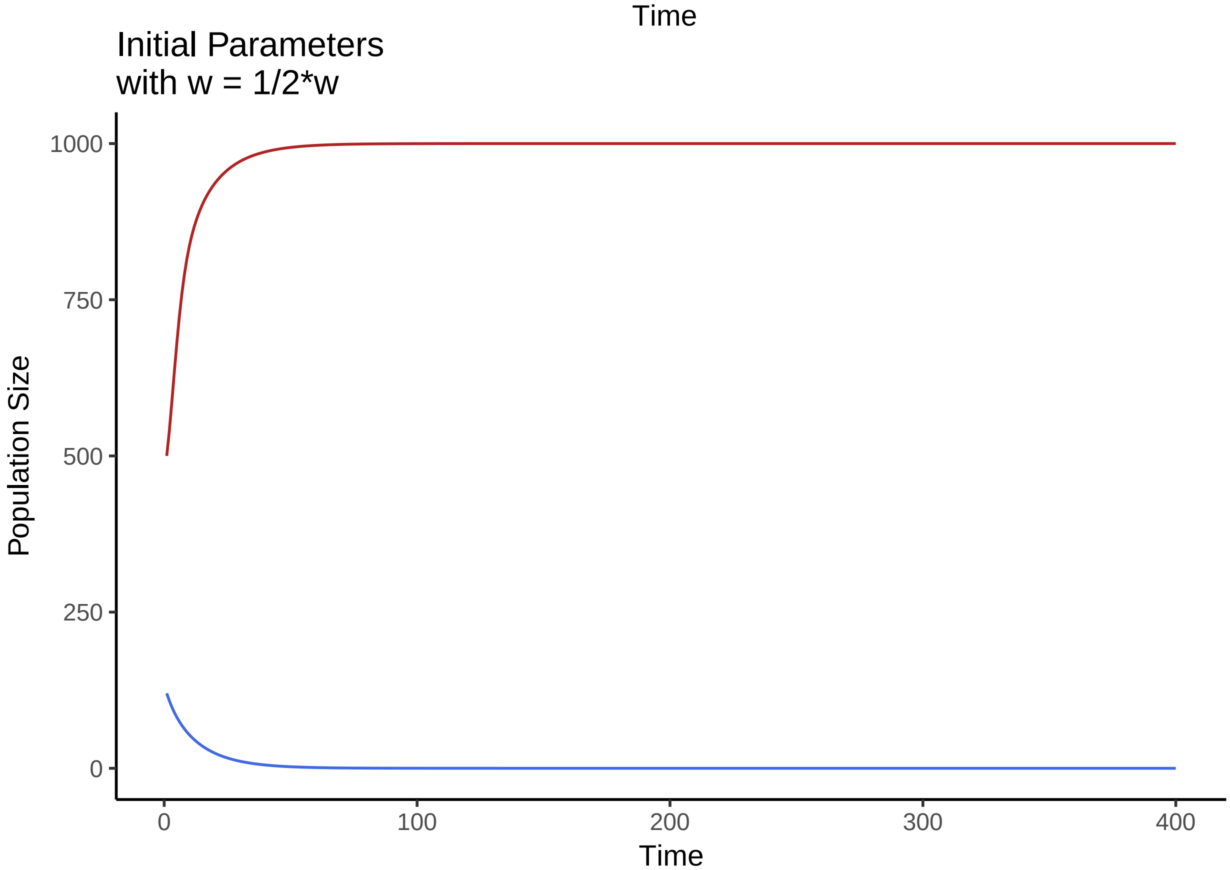
Predators are able to survive

High attack rate implies that prey will be driven to a small population. Predators can't survive if prey population is small, so they decreases

As predator population decreases, prey population is able to increase. The cycle repeats

Long-term behavior is oscillation


Initial Parameters
with w = 2*w
Population Size
0
250
500
750
0
100
200
300
400
Time


Initial Parameters
with w = 1/2*w
Population Size
0
250
500
750
1000
0
100
200
300
400
Time

Prey Denisty to Maximize Predator Attack

Low d values

Predator population can maximize attack rate more easily

Maximizing attack rate drives prey population down

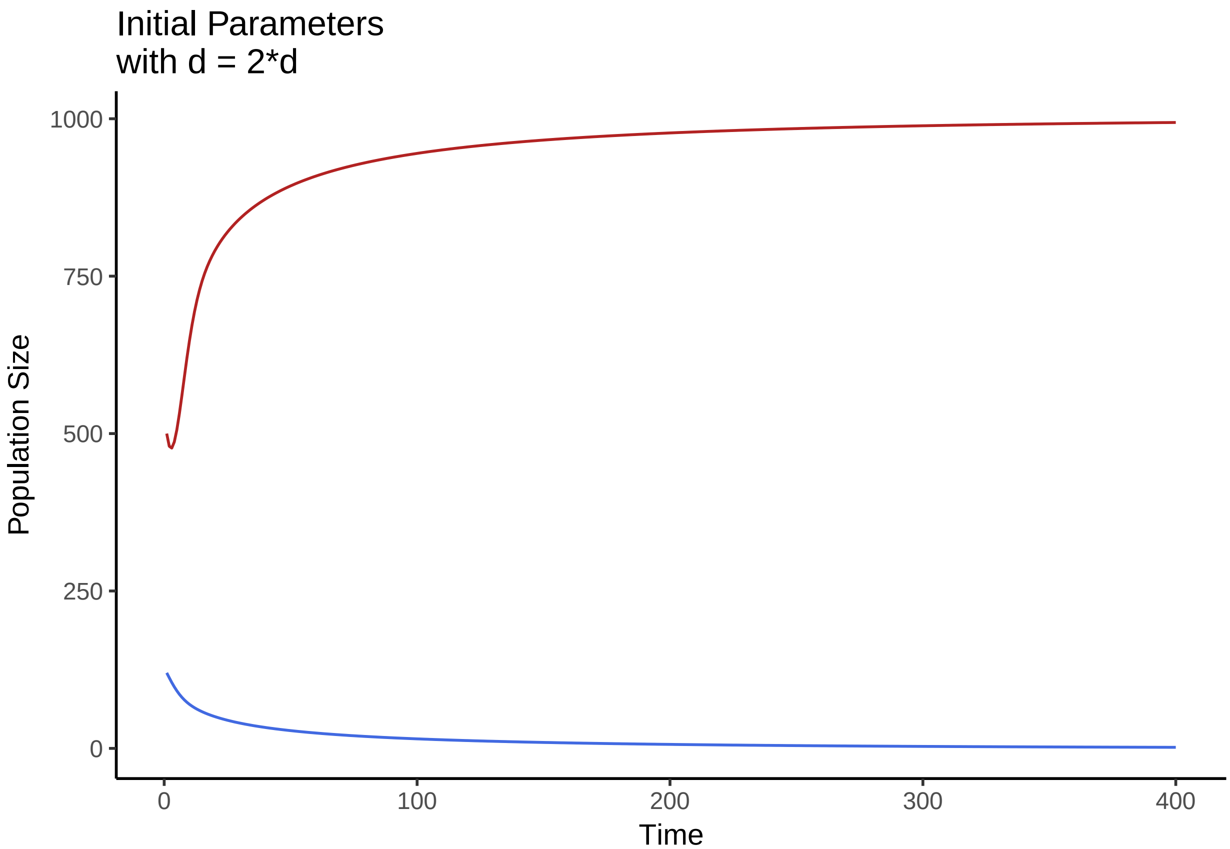
Small prey population hurts the predators, so predators decrease

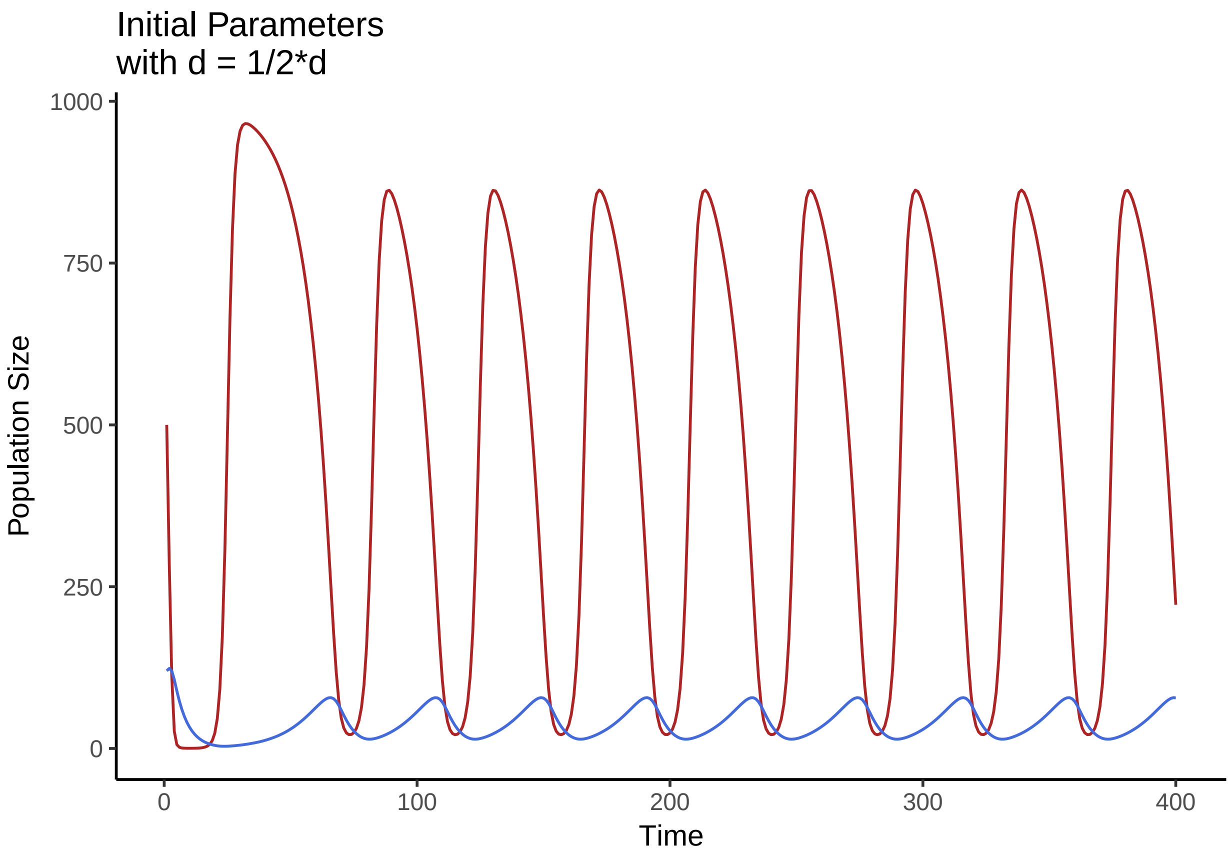
The system oscillates

High d values

Predators cannot reach an attack rate that sustains their population

Predator population goes to zero; prey go to carrying capacity





Carrying Capacity/α ($\alpha = \frac{1}{k}$)

Low α

Low alpha implies higher carrying capacity of prey

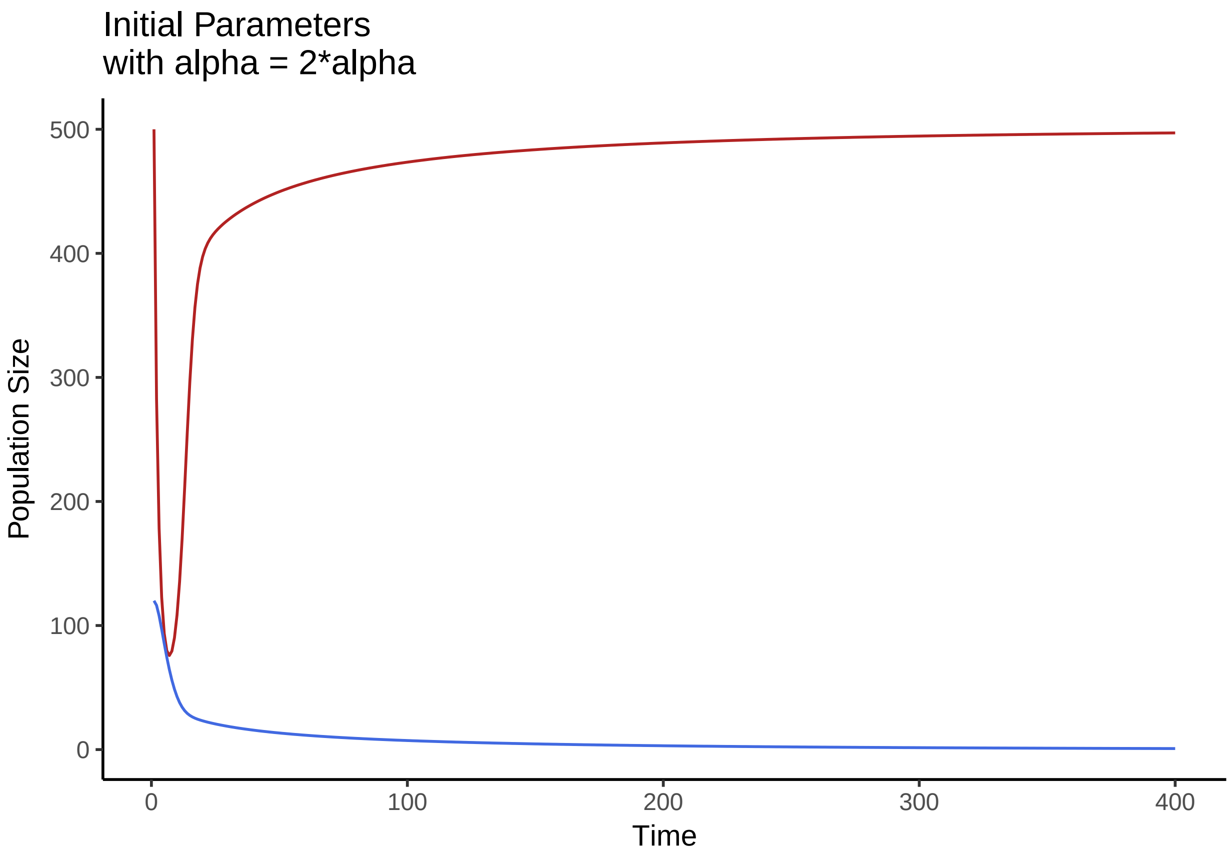
Higher carrying capacity implies higher population; this sustains predators

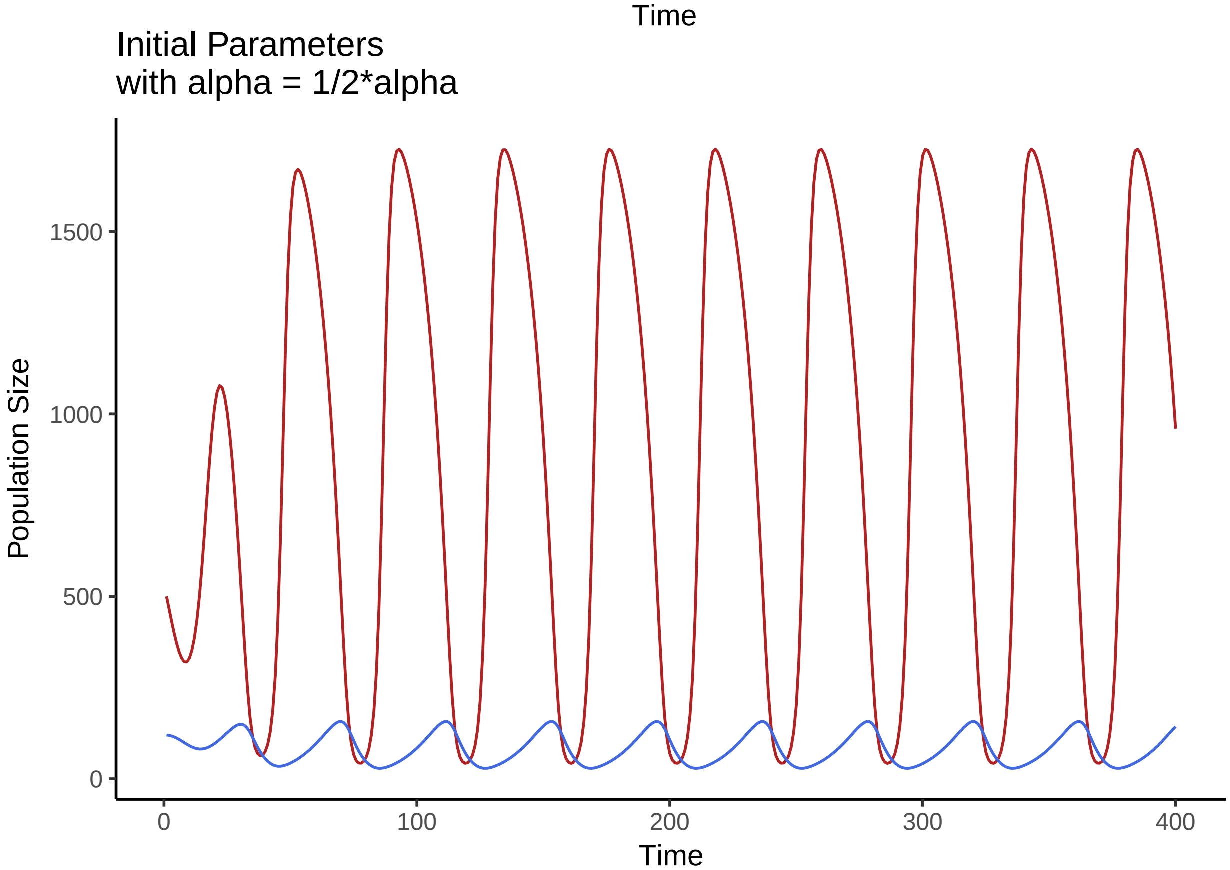
System oscillates

High α

Carrying Capacity is lower, and predators cannot sustain themseleves when there are less prey

Prey go to their carrying capacity


Initial Parameters
with alpha = 2*alpha
Population Size
500
400
300
200
100
0
0
100
200
300
400
Time


Initial Parameters
with alpha = 1/2*alpha
Population Size
1500
1000
500
0
0
100
200
300
400
Time

Prey Birthrate

At low birth rate (0.2-1.6)

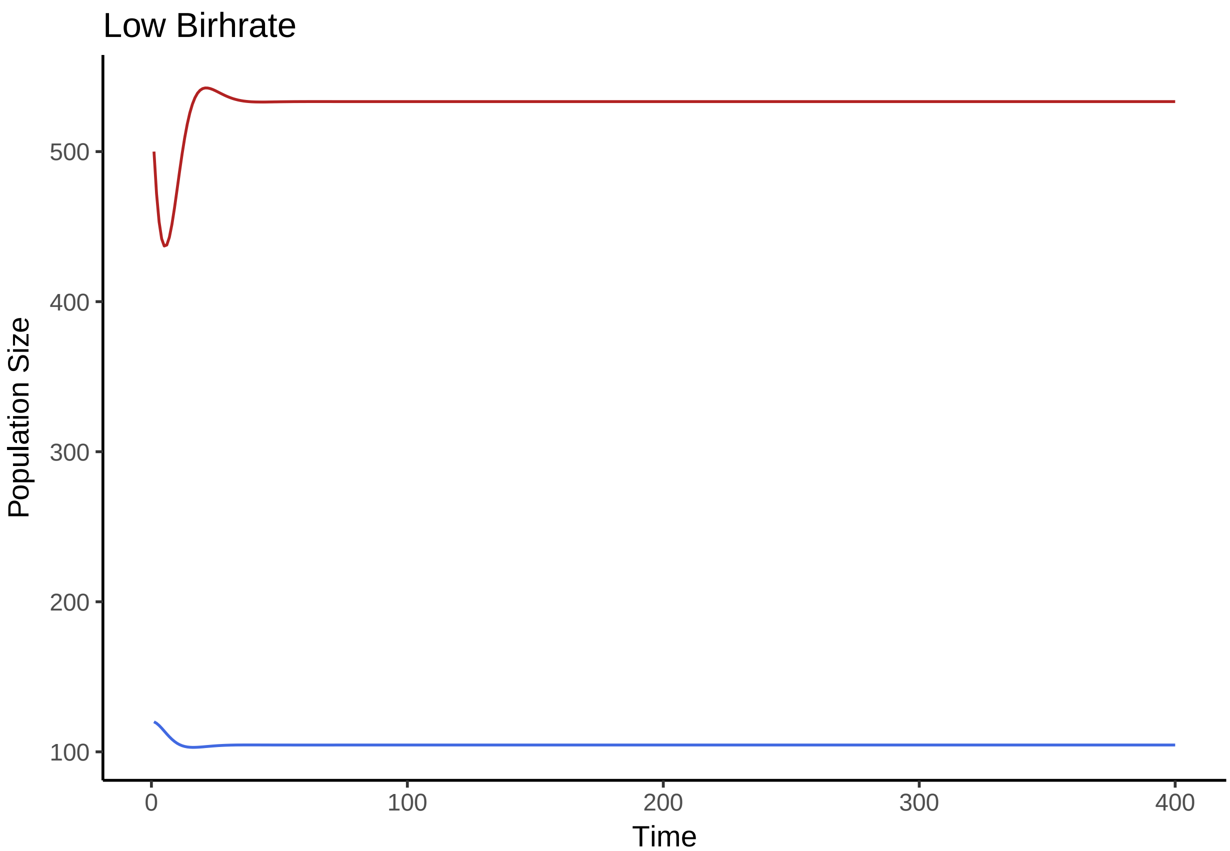
Prey and Predator population will decrease initially before increase in prey then increase in predator to equilibrium.

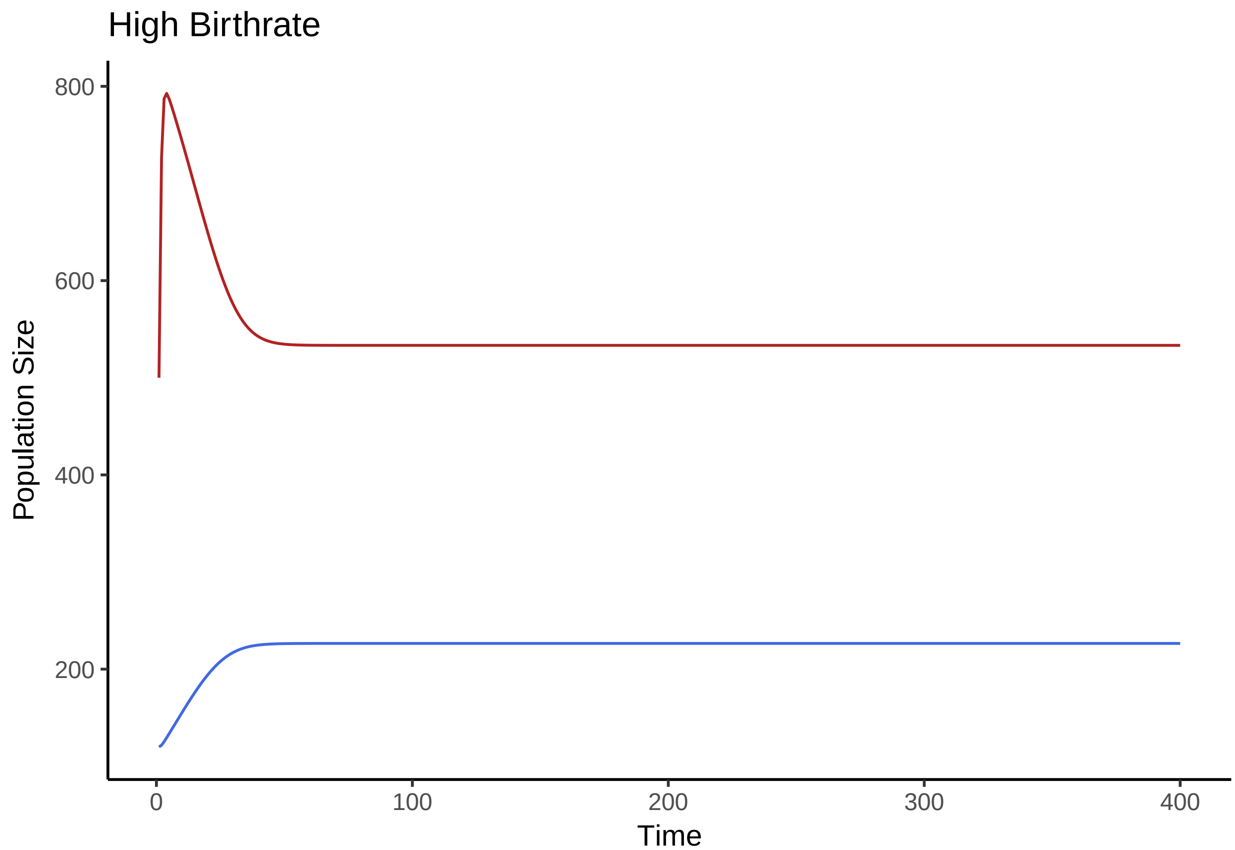
Predator death due to starvation allowing Prey population to increase to self-limiting density. Increase in Prey population allows Predator population to increase to capacity while Prey population decreases to equilibrium

At high birth rate (2.0-3.2)

Prey birthrate increases to self-limiting density before Predator decreases Prey rate to equilibrium

Initial Predator population does not require as much Prey to sustain itself, therefore Prey population reaches self-limiting density. Predator population will continue to increase until capacity is reached. Capacity for Prey and Predator will increase as birth rate of prey increases





Conversion efficiency of Prey to Predator

Low efficiency conversion (0.0175-0.0525)

Rate of Prey will initially decrease because Predator is present in system.

Rate of Predator decreases because of efficiency to convert energy to more Predator, the population will decrease to 0 as Prey population increases to self-limiting density. Rate to capacity determined by other parameters in the function

Medium efficiency conversion and mutual equilibrium (0.07-0.0875)

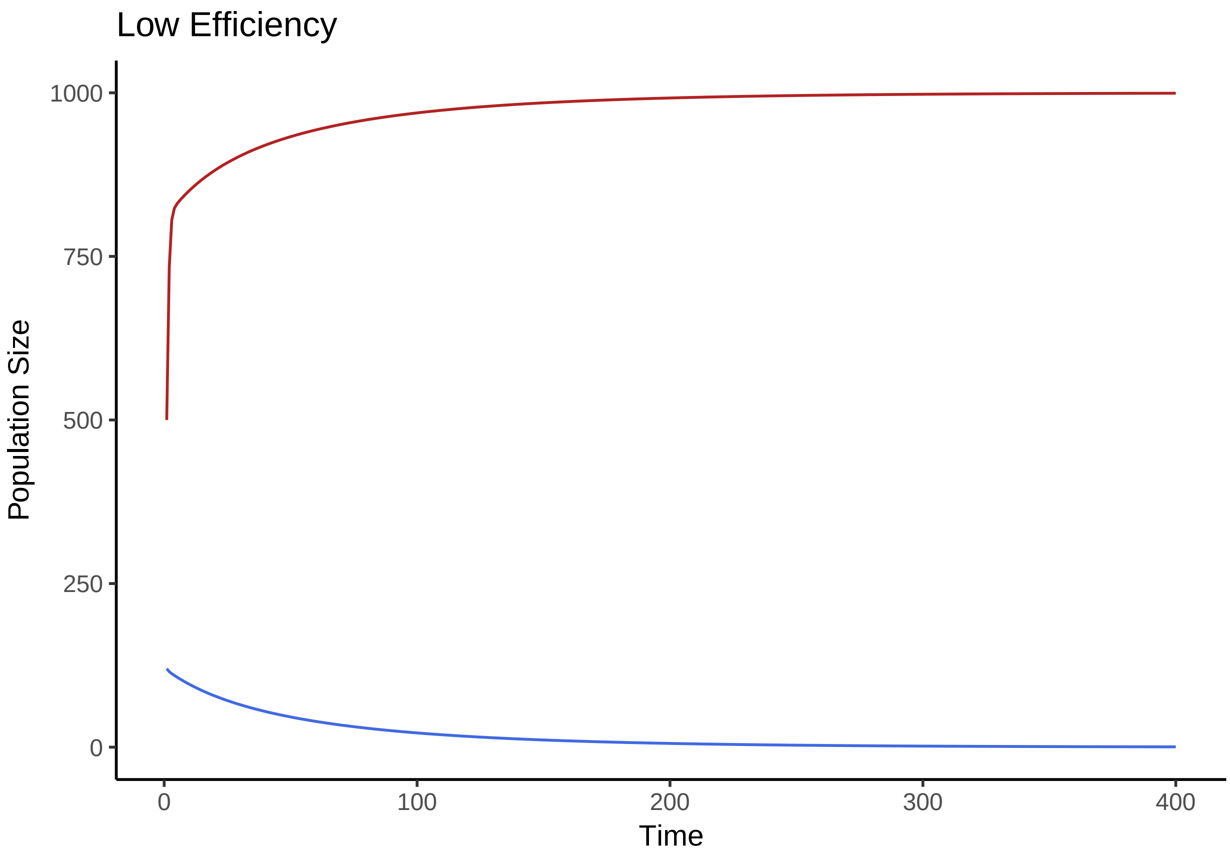
Rate of Prey and Predator will oscillate.

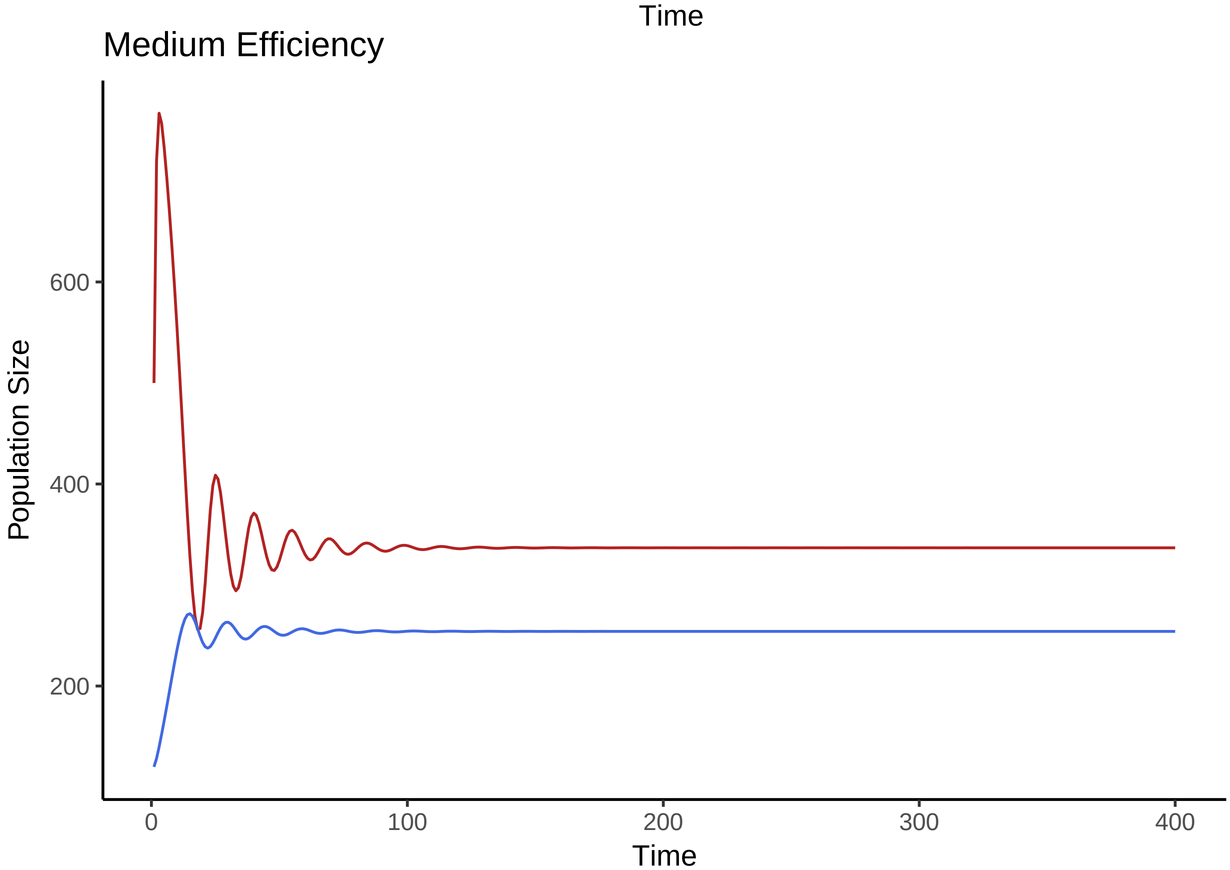
Prey population drops as Predator starts consuming. Predator reaches a population size that is not sustained by the amount of Prey and population will decrease. Prey will bounce back and cycle will begin again until reaching a stable equilibrium.

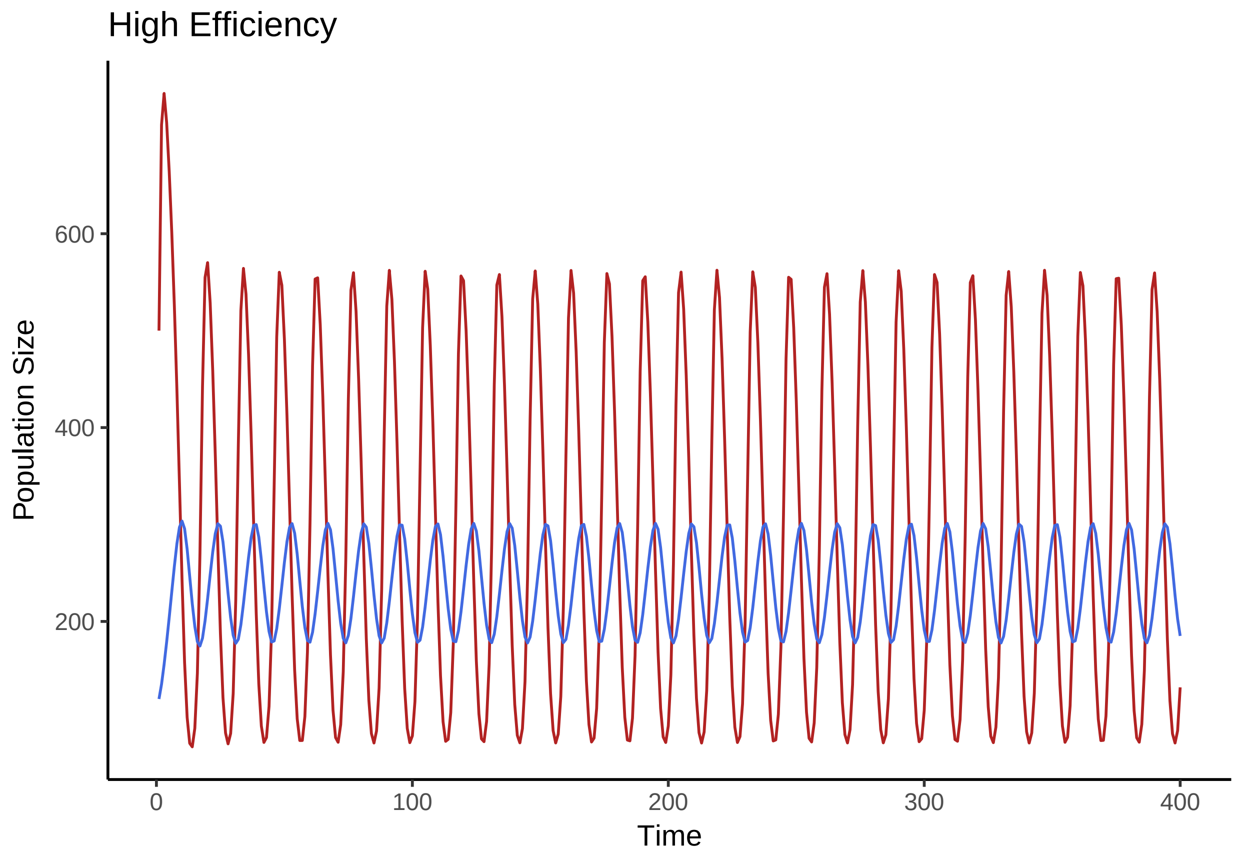
High efficiency conversion (0.1050-0.28)

Rate of exchange from Prey to Predator occurs too quickly.

System will change from decreasing Prey and increasing Predator until Predator density is reached. Predators will decrease from starvation or lack of Prey to support system until Prey numbers bounch back. Increase in efficiency will increase maximum capacity of predator


Low Efficiency
Population Size
1000
750
500
250
0
0
100
200
300
400
Time


Medium Efficiency
Population Size
600
400
200
0
100
200
300
400
Time



Predator Death Rate

Low Death Rate (0.05-0.2)

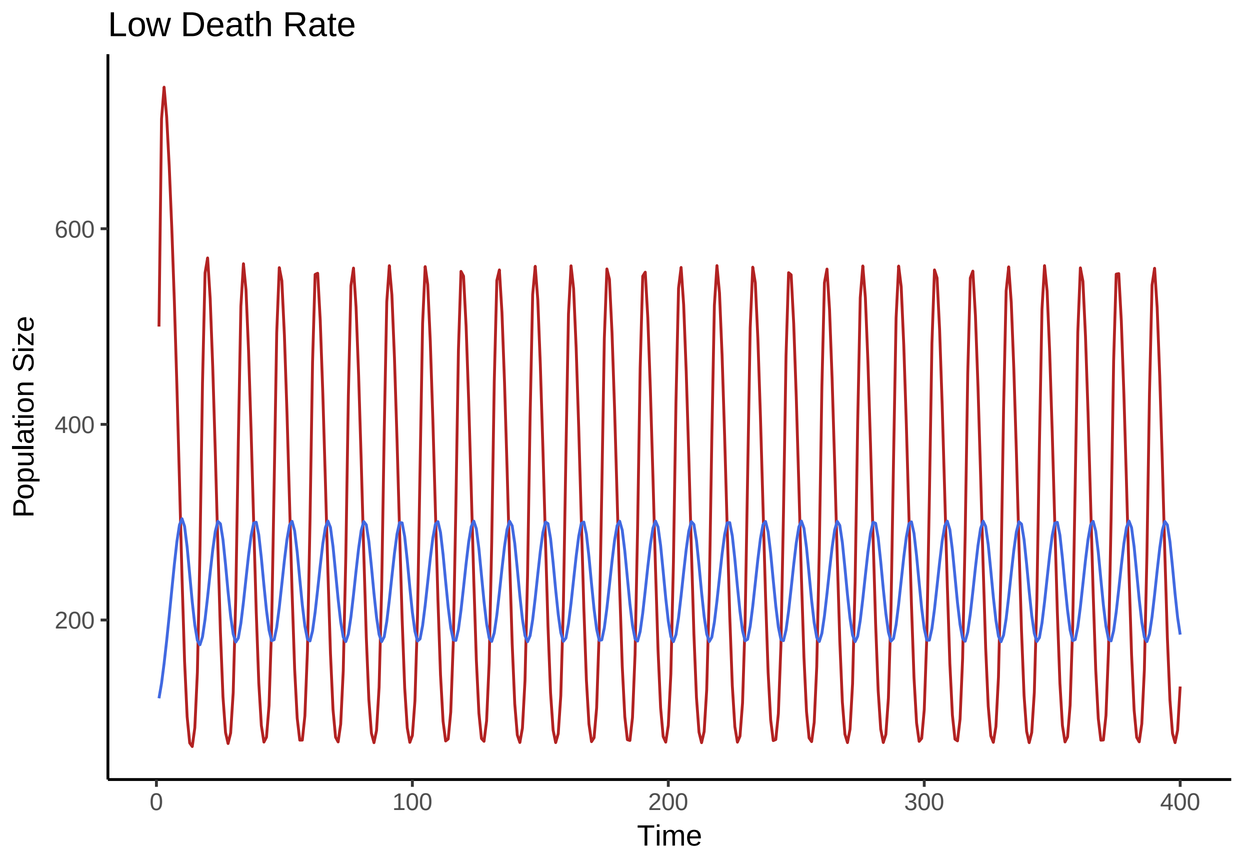
Low death rate leads to depletion of Prey population

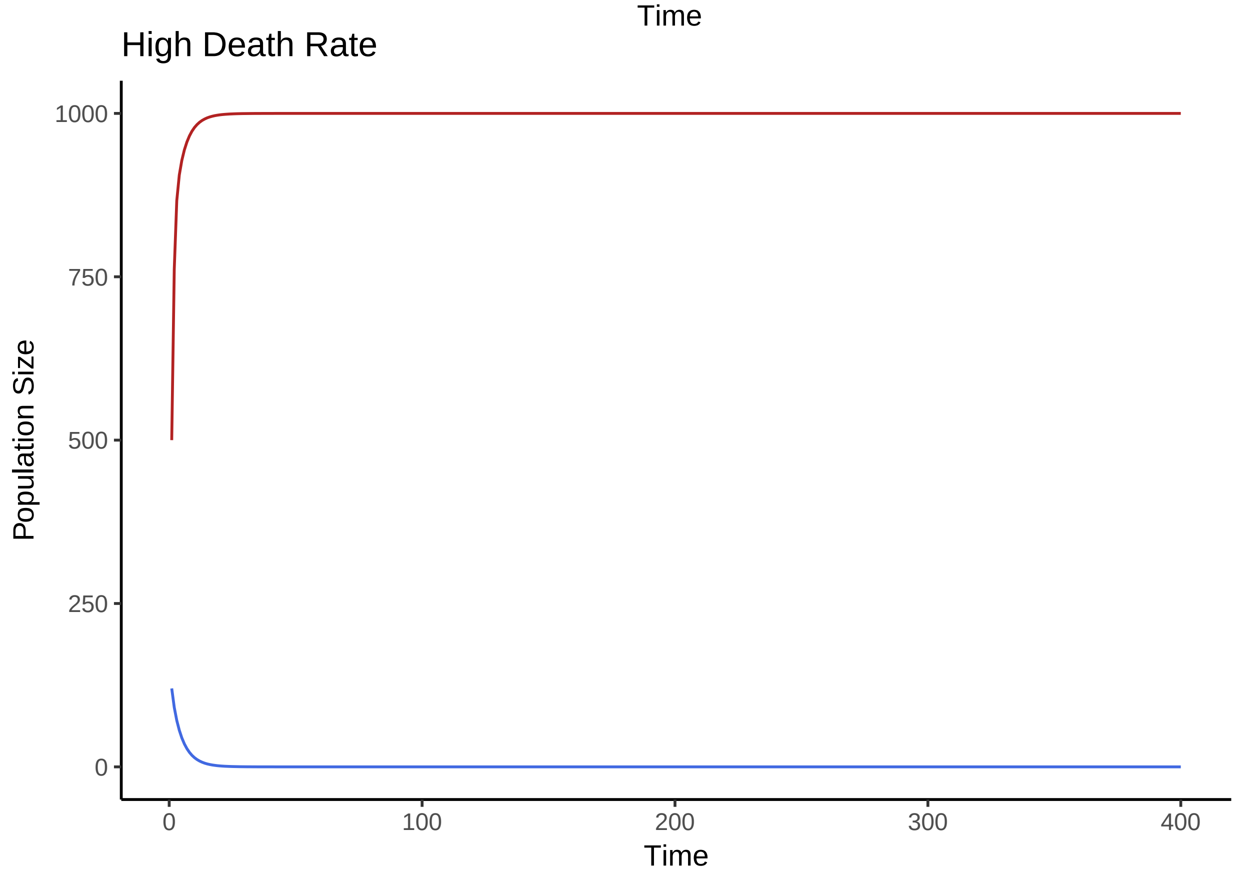
Decrease in Prey population decreases Predator density limit. Oscillation of both population until equilibrium is reached, or will continue to oscillate between capacity and 0

High Death Rate (0.25-0.8)

Initial dip in Prey population results from initial Predator population

High death rate results in eradication of Predator and increase of Prey Population to self-limiting density


Low Death Rate
Population Size
600
400
200
0
100
200
300
400
Time


High Death Rate
Population Size
1000
750
500
250
0
0
100
200
300
400
Time

Questions

How do the dynamics differ from Lotka-Volterra?

Dynamics in R-M models allow for a stable equilibrium population. The L-V population consistantly oscillates between Prey depletion and Predator starvation to Prey. In the absence of predators, Prey in the L-V model grow exponentially, while Prey in the R-M model grow logistically.

What can you say about the role of each parameter, especially what causes the dyamics to differ between the L-V and R-M models?

The extra parameters seen in the R-M model adds complexity that allows for both populations to reach a population size in which co-habitation can be possible within some range of the parameters.

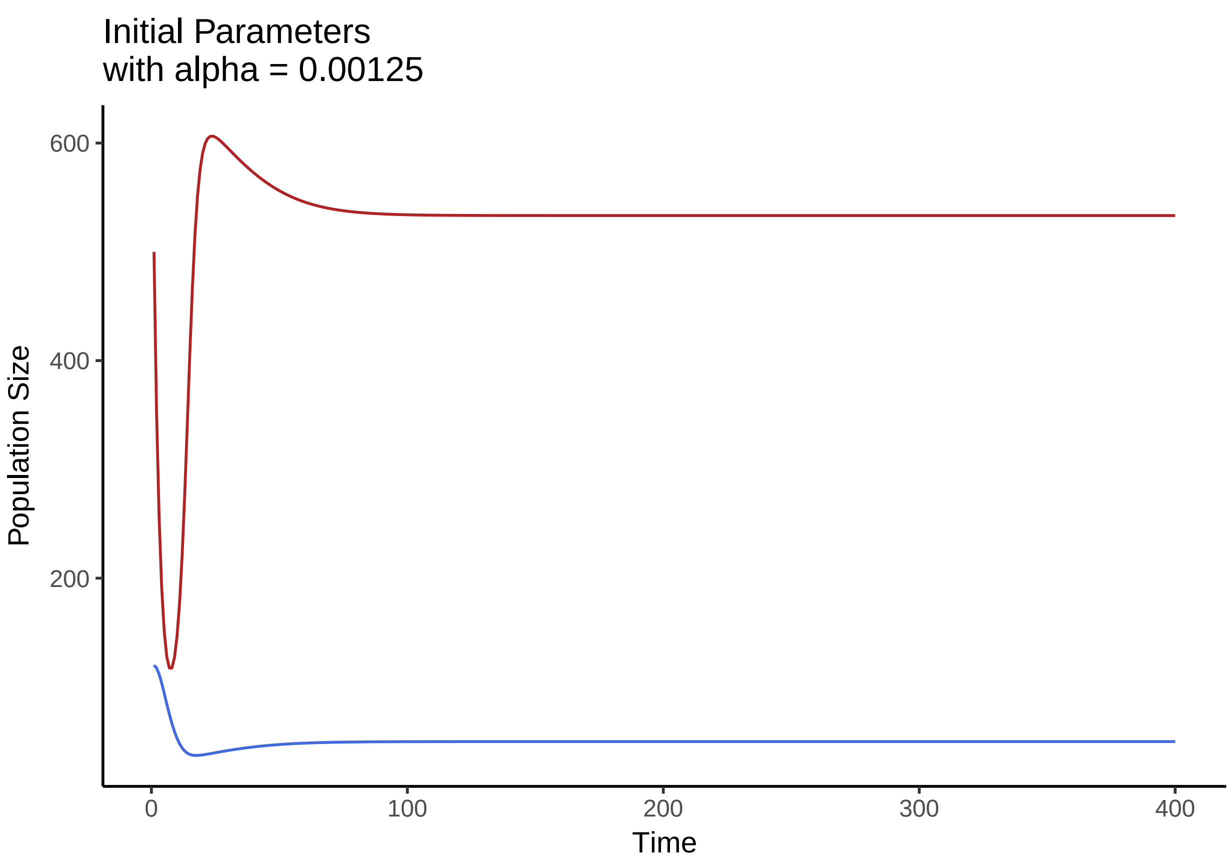
What is the relationship between parameter values and predator abundance?

Predator abundance has a higher capacity when Prey birthrate (b) is higher, Prey carrying capacity is higher (lower α), Prey-to-Predator conversion (e) is higher, Preadator attack rate (w) is higher, Prey population to maximize predator attack rate (d) is lower, and Predator death rate (s) is lower.

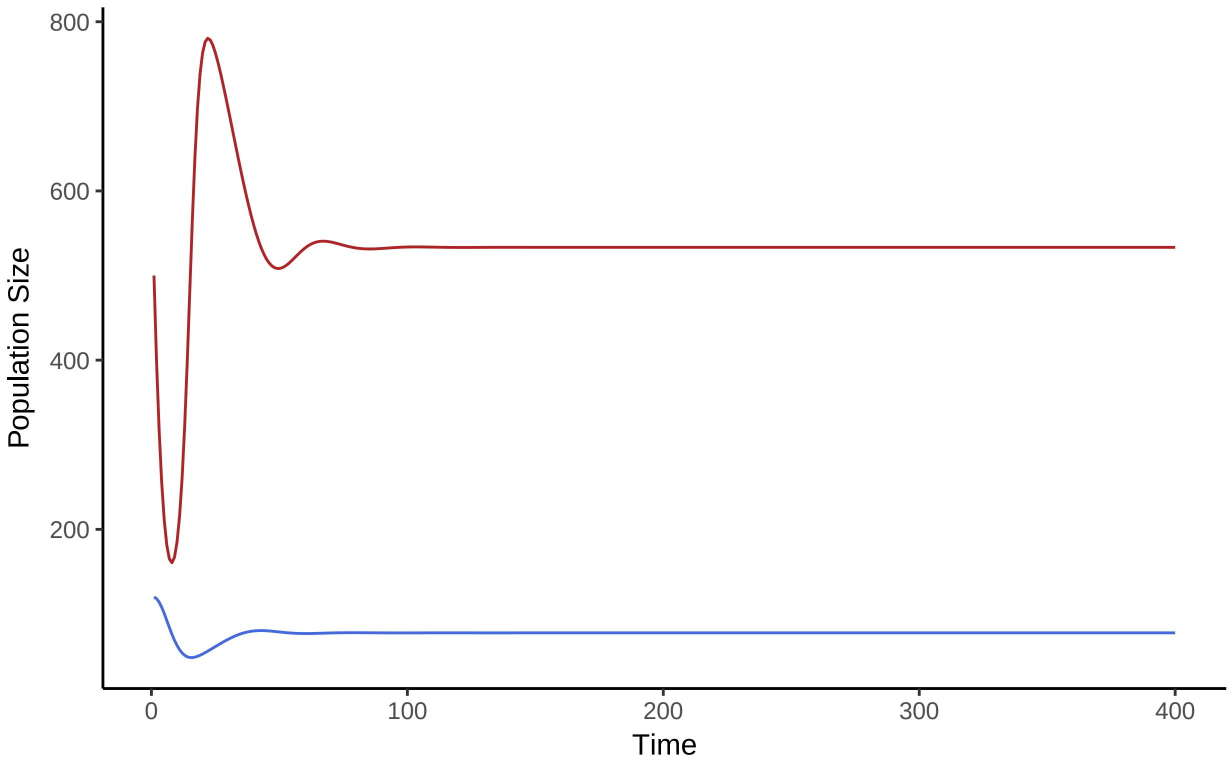
Paradox of Enrichment

Conceptually, higher carrying capacity (lower α) should lead to predator stability because more prey will buffer the population from extinction.

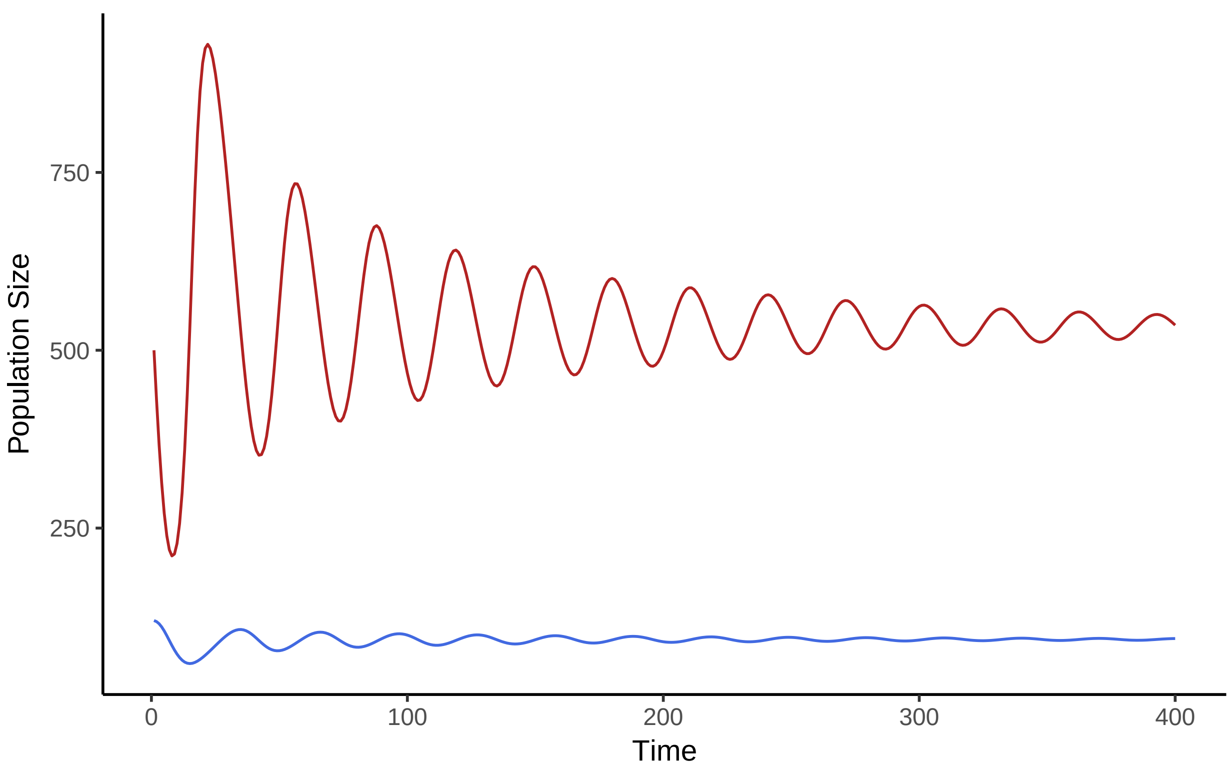
However, the RM model shows that there is a negative relationship between prey carrying capacity and predator stability: as alpha decreases (carrying capacity increases), the population of predators becomes less stable. This is termed the *Paradox of Enrichment*. Let's see some examples of this:

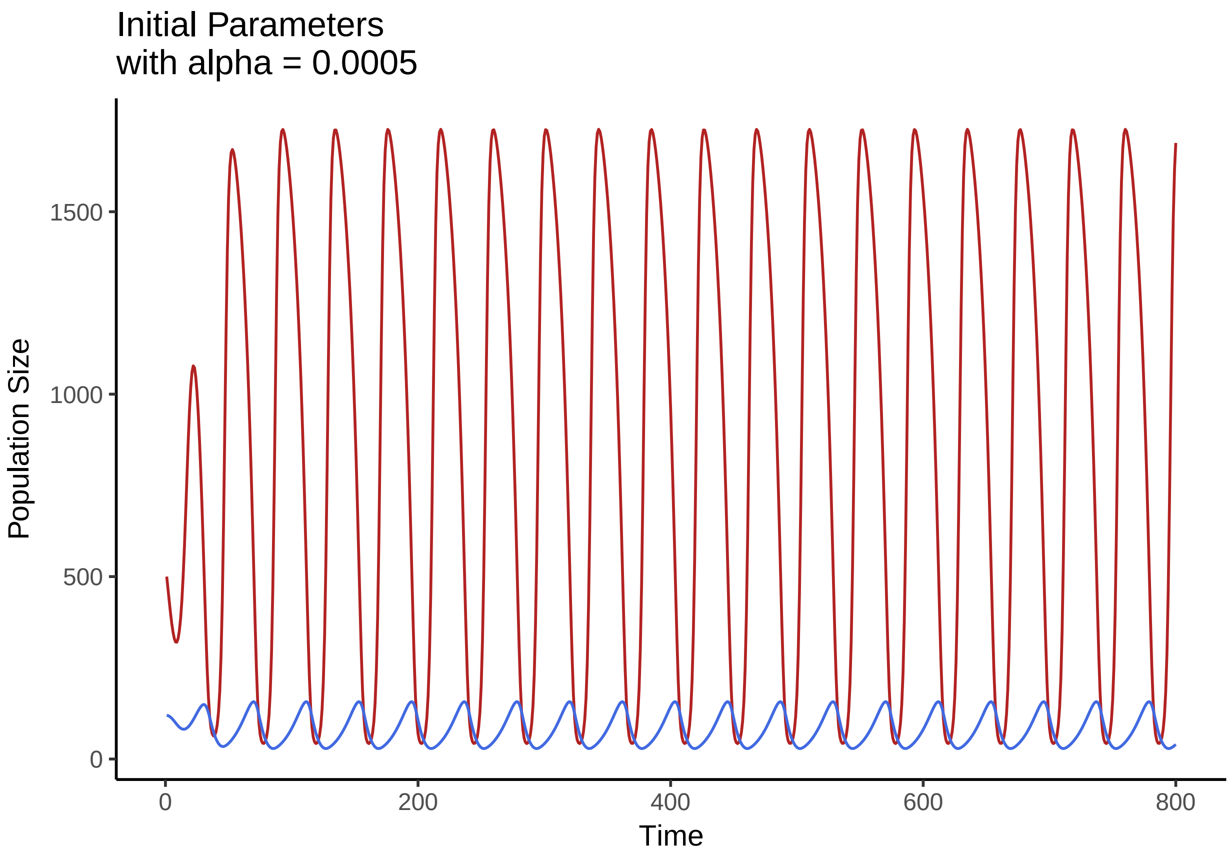


Initial Parameters
with alpha = 0.0009



Initial Parameters
with alpha = 0.0007





At low carrying capacities, the self-limiting aspect of the prey population leads to a stable equilibrium between predators and prey. However, as carrying capacity increases, a certain stability threshold is passed. Prey population is able to grow to such a point that predators can eat enough prey, driving the prey population down, which in turn drives the predator population down, leading to oscillation.