# Stability

Alex Reed

2023-06-09

Consider how you might add hunting of prey to the predator prey model that we've been using in class

Build this model (e.g add harvesting to the lotvmodK.R), you should make sure that you don't hunt more prey than exist. To ensure that you might also add a minumum prey population input that must be met before hunting is allowed.

```
#source Lot. Voltera Model with Hunting
source("lotvmodK_hunt.R")
```

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)

Assume the following rprey=0.95, alpha=0.01, eff=0.6,pmort=0.4, K=2000

```
# parameters and time periods
pars = data.frame(rprey=0.95, alpha=0.01, eff=0.6, pmort=0.4, K=2000)
# 50 year time-period
time_period = seq(from=1, to=500, by  = .1)

# range of hunting rates and minimum prey populations
hunting_rates = c(0.05, 0.1, 0.8)
min_prey_pop = c(0, 200, 1000)

# initial population
currpop = c(prey=500, pred=100)

results = list()

# perform simulations with different hunting rates and minimum prey populations
for (h in hunting_rates) {
    for (min_prey in min_prey_pop) {
        pars$h = h
        pars$min_prey = min_prey
        res = ode(func=lotvmodK_hunt, y=currpop, times=time_period, parms=pars)
        results[[paste(h, min_prey, sep="_")]] = as.data.frame(res)
    }
}
```

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

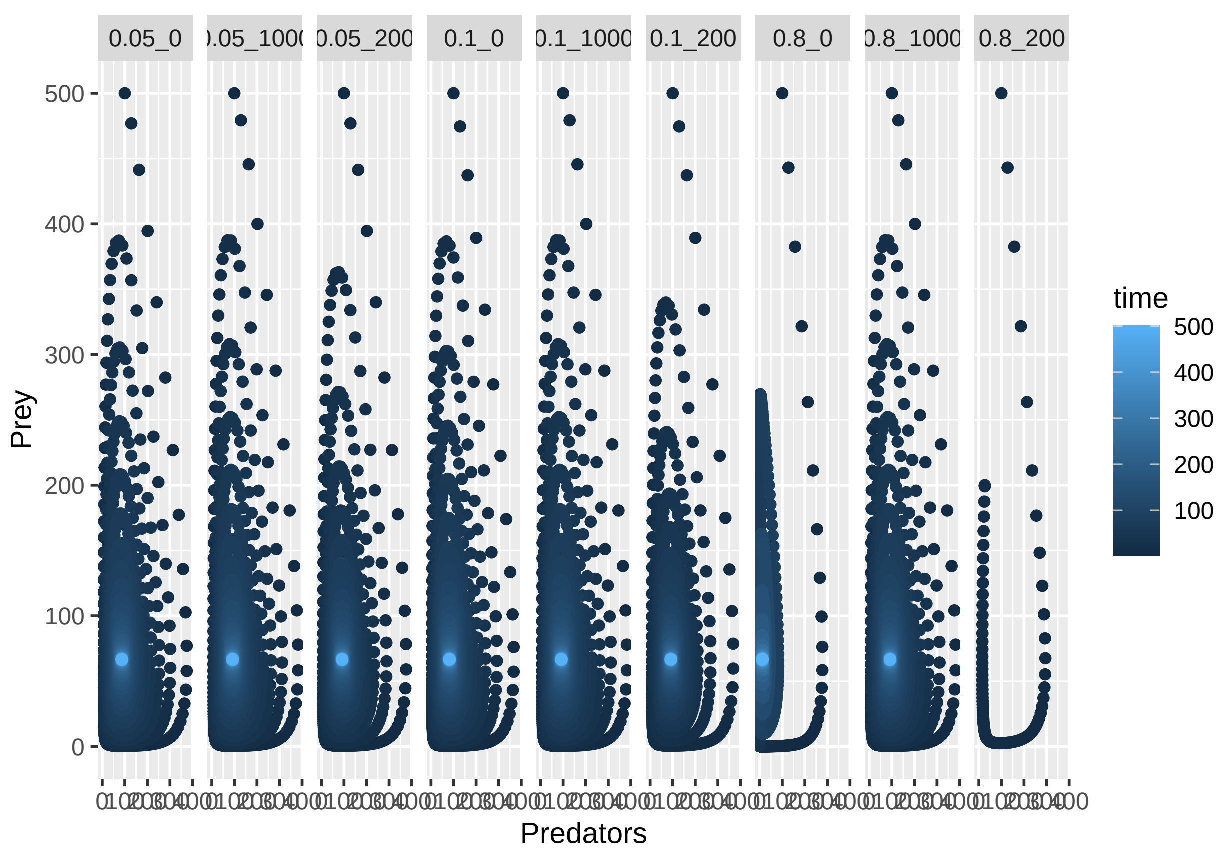
```
##
## In above message, R1 = 13.0086
##

## Warning in lsoda(y, times, func, parms, ...): an excessive amount of work (>
## maxsteps ) was done, but integration was not successful - increase maxsteps

## Warning in lsoda(y, times, func, parms, ...): Returning early. Results are
## accurate, as far as they go
```

```
# combine results and create a dataframe with columns for hunting rate and minimum prey population
df <- bind_rows(results, .id = "id") %>%
  mutate(hunting_rate = as.factor(str_split(id, "_", simplify = TRUE)[,1]),
         min_prey = as.numeric(str_split(id, "_", simplify = TRUE)[,2]))

# predator and prey graphs for the different hunting and pop rates
ggplot(as.data.frame(df), aes(pred, prey, col=time)) +
  geom_point() +
  labs(y="Prey",x="Predators") +
  facet_grid(~id)
```




A key challenge is how you might want to define stability? Its up to you but you will need to write a sentence to explain why you chose the measure that you did. It could be something as simple as maintaining a population above some value 50 years into the future.

**I am defining stability as having both the predator and prey above 60 for 50 years. I picked a threshold of 60 to avoid the risk of extinction of my predator and prey.**

```r
stable_df <- list()

# results that meet stability definition
for (name in names(results)) {
  df <- results[[name]]
  if (nrow(df) >= 500) {
    final_pops <- df[(nrow(df)-499):nrow(df), ]
    if (all(final_pops$prey >= 60) && all(final_pops$pred >= 60) && nrow(final_pops) == 500) {
      df$hunting_min <- name  # Add hunting_min column
      stable_df[[name]] <- df
    }
  }
}

# combine the stable populations into a single dataframe
stable <- bind_rows(stable_df, .id = "hunting_min")

# stable populations
unique(stable$hunting_min)
```

```
## [1] "0.05_0"    "0.05_200"  "0.05_1000" "0.1_0"     "0.1_200"   "0.1_1000"
## [7] "0.8_1000"
```

Recommendation: **It appears that with a hunting rate of 0.8, you would need a high number of prey (at least 1000) to keep the system stable. Lower hunting rates will stay stable with fewer initial prey populations. A sustainable hunting strategy would be to maintain a hunting rate of 0.1 or below. This rate provides a balance that allows for hunting while also ensuring the stability of the system.**