Hunting

Hunting Assignment

```r
library(sensitivity)
library(here)
library(tidyverse)
library(deSolve)
```

Part 1

Include hunting in function:

```r
source("R/hunting.R")
```

Part 2

Try different hunting values: which ones give *stable* prey population?

Stability: We define stability as no fluctuations in the prey population after 1 year. Graphically, the prey population will flatten out. We compared the graphs from 6 different scenarios to determind which vales for `min_prey_hunt` and `hunt_rate` created stability.

Initial conditions:

```r
# note the use of with
# initial conditions
currpop <- c(prey = 1000, pred = 10)

# 2 years
years <- seq(from = 1, to = 2*365, by = 1)
```

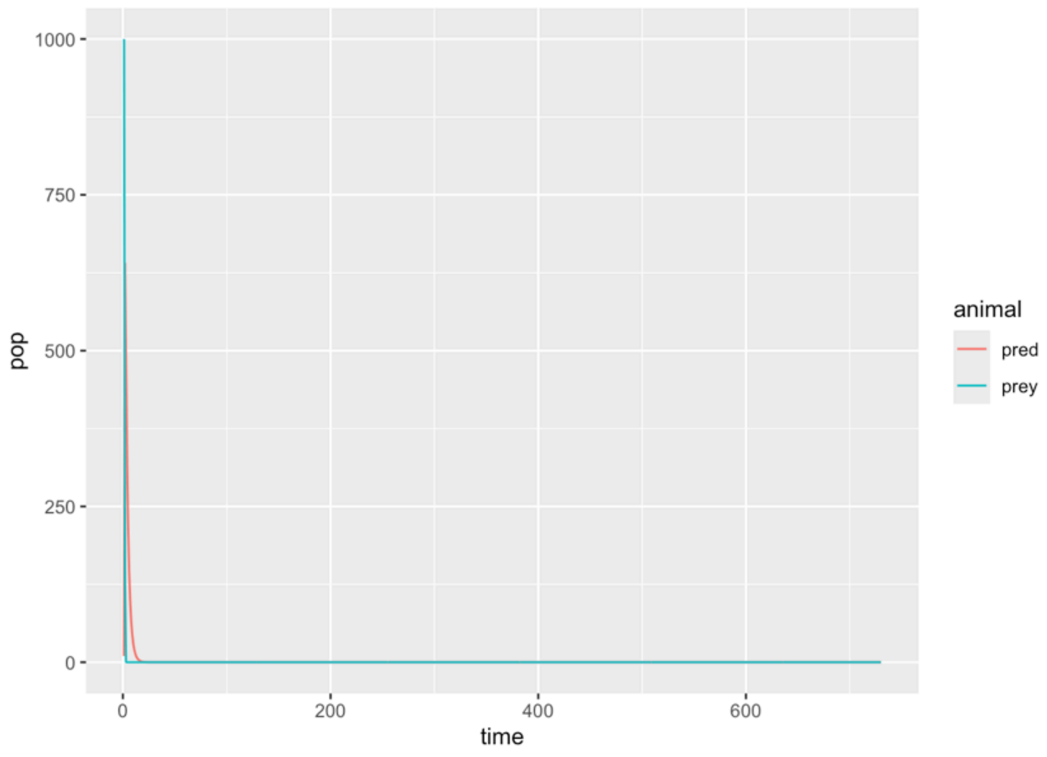
Build function to test scenarios:

```r
test_scenarios <- function(min_prey_hunt, hunt_rate) {
  # set parameters for scenario 1
pars <- c(rprey = 0.95, alpha = 0.01, eff = 0.6, pmort = 0.4, K

# run the model
res <- ode(func = hunting, y = currpop, times = years, parms = 

# graph the results
head(res)

# rearrange for easy plotting
resl <- as.data.frame(res) %>% pivot_longer(-time, names_to = "
plot <- ggplot(resl, aes(time, pop, col = animal)) +
  geom_line()

return(plot)

}
```

Scenario 1:

- `min_prey_hunt`: 0
- `hunt_rate`: 50

```
test_scenarios(0, 50)
```

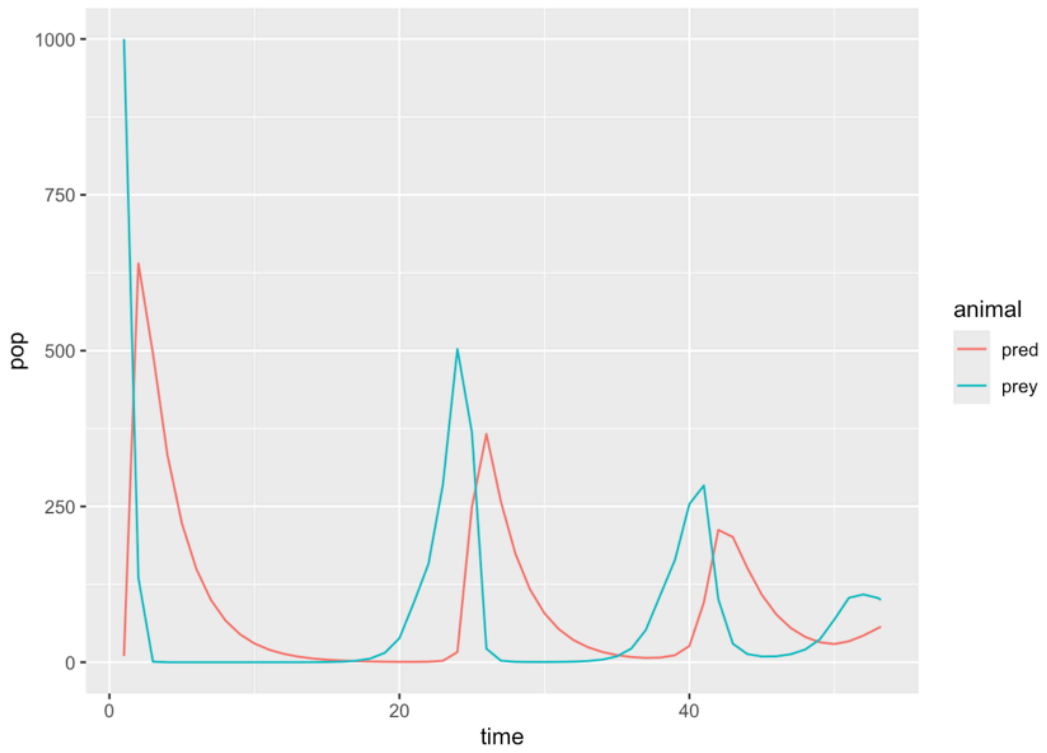


Scenario 2:

- `min_prey_hunt`: 100
- `hunt_rate`: 50

```
test_scenarios(100, 50)
```

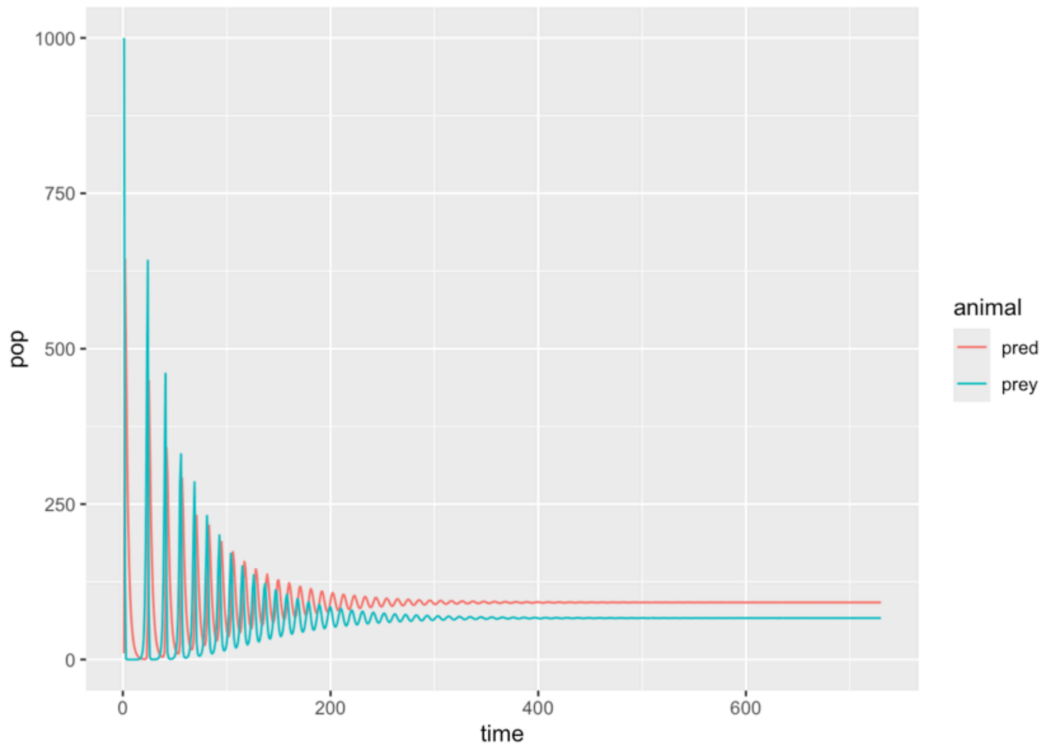
```
DLSODA-  At current T (=R1), MXSTEP (=I1) steps
      taken on this call before reaching TOUT
In above message, I1 = 5000

In above message, R1 = 53.2125


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
```



Scenario 3:

- `min_prey_hunt`: 500
- `hunt_rate`: 50

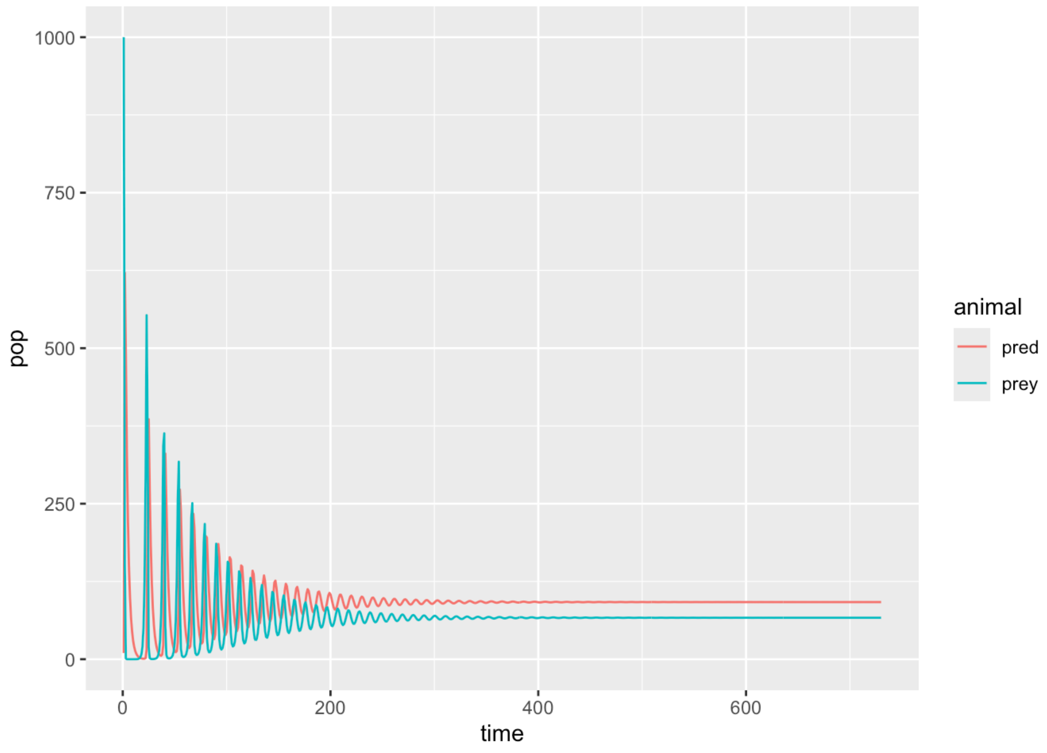
```r
test_scenarios(500, 50)
```



Scenario 4:

- `min_prey_hunt`: 500
- `hunt_rate`: 100

```r
test_scenarios(500, 100)
```

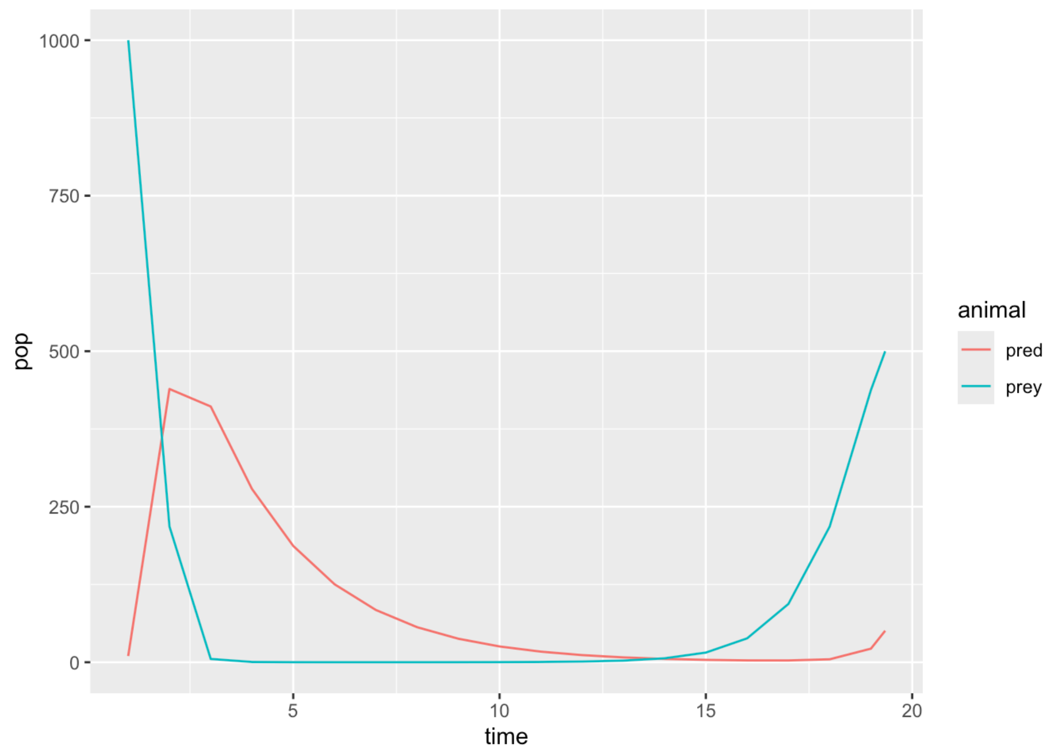


Scenario 5:

- `min_prey_hunt`: 500
- `hunt_rate`: 500

```
test_scenarios(500, 500)
```

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

In above message, R1 = 19.3453


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
```



Prey population *stabilizes* after about a year with Scenarios 3 an 4 (`min_prey_hunt` = 500, `hunt_rate` = 50 and `min_prey_hunt` = 500, `hunt_rate` = 100). This shows that `min_prey_hunt` performs best around 25% of carrying capacity, and `hunt_rate` performs well at 10-20% of `min_prey_hunt`.