Hunting Assignment

AUTHOR
Tom Gibbens-Matsuyama

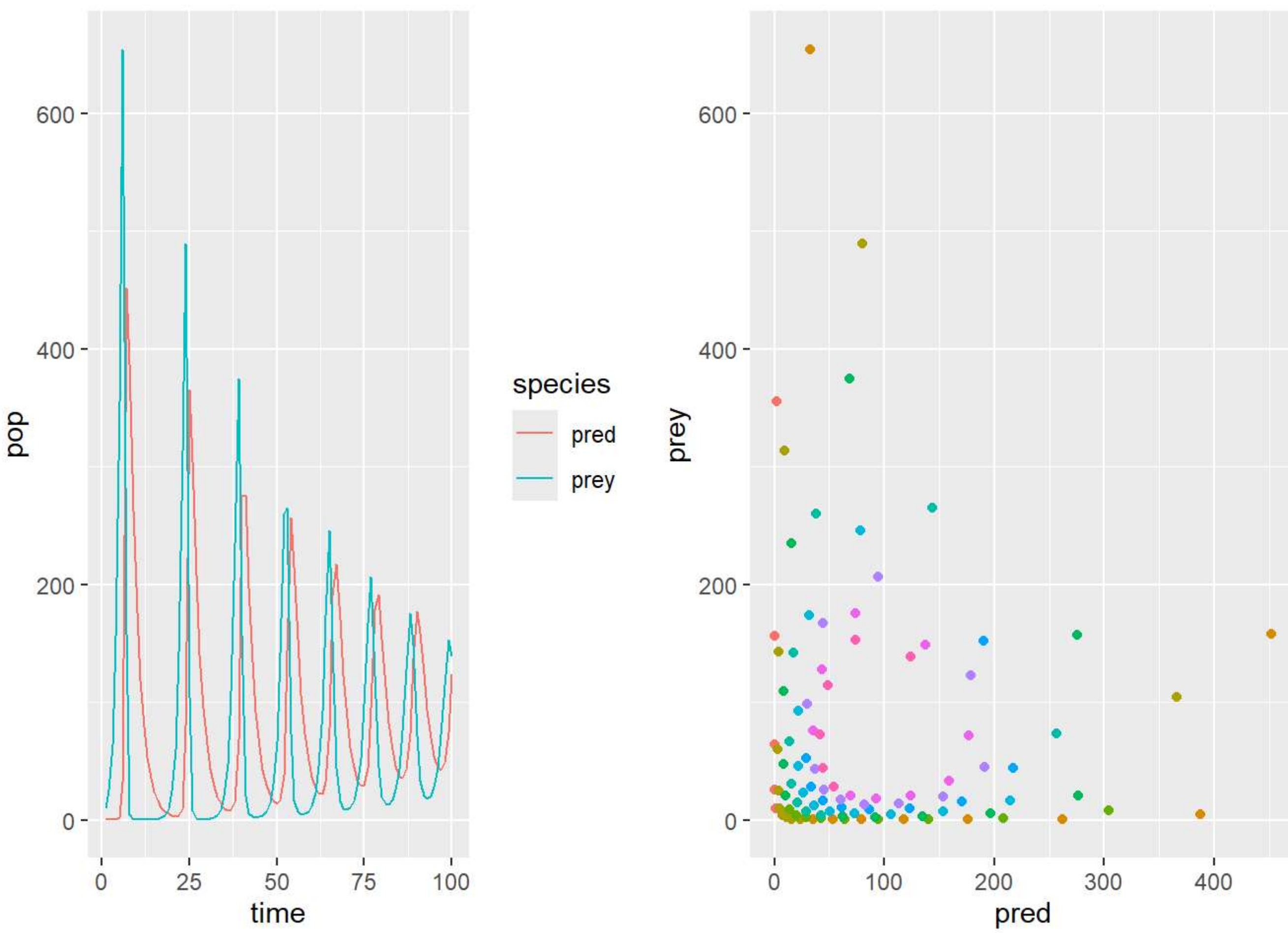
```
library(deSolve)
library(ggpubr)
library(tidyverse)
library(sensitivity)
library(lhs)
library(purrr)
library(ggpubr)
library(here)
```

Model without hunting for given parameters

- rprey = 0.95
- alpha = 0.01
- eff = 0.6
- pmort = 0.4
- K = 2000

```
source(here("R/lotvmodK.R"))

# initial conditions
currpop <- c(prey = 10, pred = 1)

# set parameter list
pars <- c(rprey = 0.95, alpha = 0.01, eff = 0.6, pmort = 0.4, K = 2000)

# times when you want to evaluate
days <- seq(from = 1, to = 100)

# run our differential equation solver
res <- ode(func = lotvmodK, y = currpop, times = days, parms = pars)

# rearrange for plotting
resl <- as.data.frame(res) %>% pivot_longer(-time, names_to = "species", values_to = "pop")

# graph both populations over time
p1 <- ggplot(resl, aes(time, pop, col = species)) +
  geom_line()


# also look at relationships between preditor and prey population and use color for time
# I will remove the legend here to make it easier to see
p2 <- ggplot(as.data.frame(res), aes(pred, prey, col = (round(time / 10)))) +
  geom_point() +
```

```
  theme(legend.position = "none")

p2 <- ggplot(as.data.frame(res), aes(pred, prey, col = as.factor(round(time / 10)))) +
  geom_point() +
  theme(legend.position = "none")

ggarrange(p1, p2)
```



Part 2

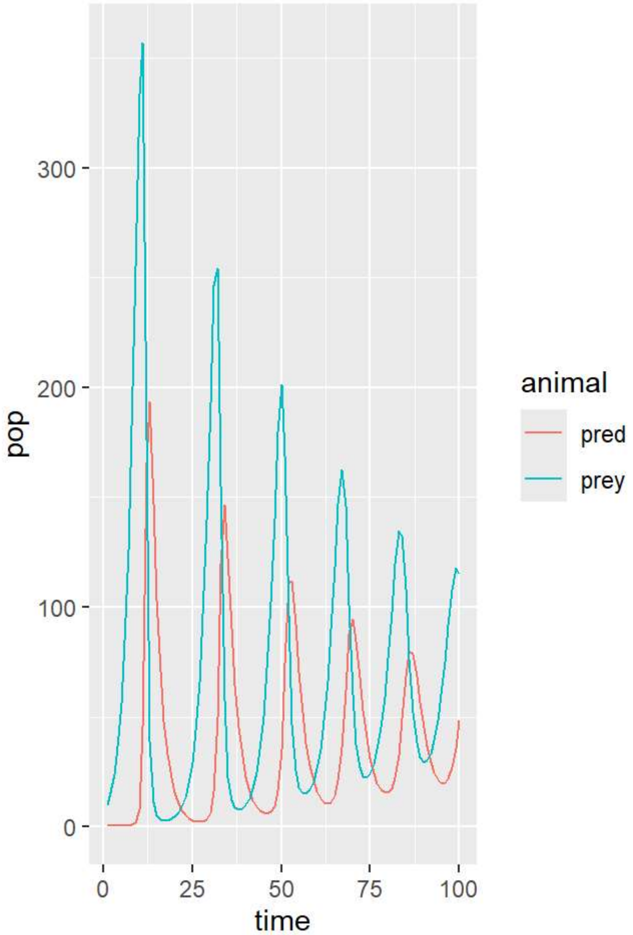
Now that we know what the model looks like for the given parameters, let's include hunting and explore how hunting affects the stability. The parameters we will adjust are: - beta - hunters - prey_thresh

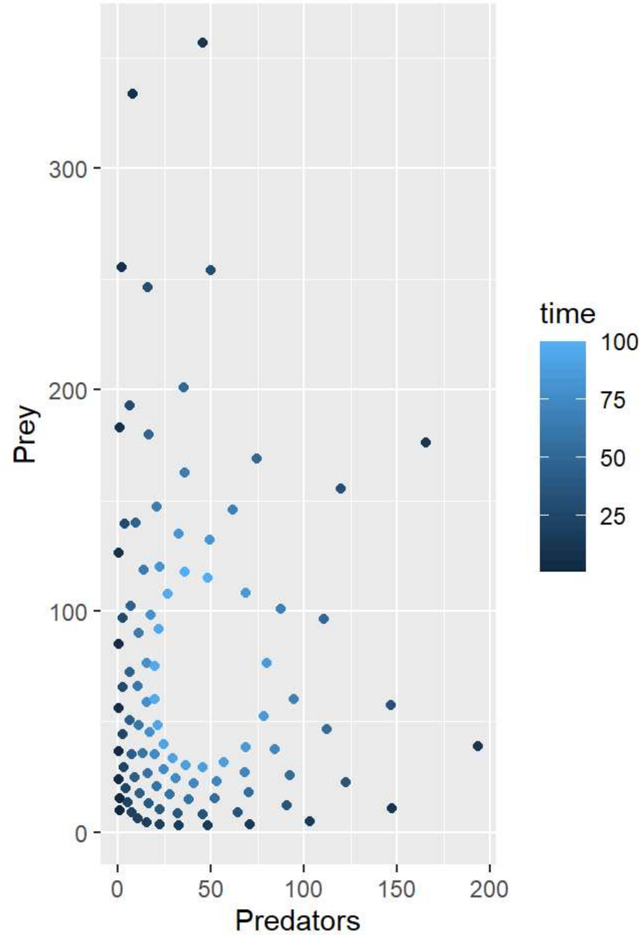
In this case, I am defining the stability of the predator prey relationship to look the same after time = 100 as in the above model (where hunting is not included).

First Model

```
source(here("R/hunting_model.R"))

# note the use of with
# initial conditions
currpop <- c(prey = 10, pred = 1)
```

```r
# time points to see results
days <- seq(from = 1, to = 100, by = 1)

# set parameters
pars <- c(rprey = 0.95, alpha = 0.01, eff = 0.6, pmort = 0.4, K = 2000, beta = 0.5, hunters = 1,

# run the model
res <- ode(func = lotvmod_hunting, y = currpop, times = days, parms = pars)

# rearrange for easy plotting
resl <- as.data.frame(res) %>% pivot_longer(-time, names_to = "animal", values_to = "pop")
p1 <- ggplot(resl, aes(time, pop, col = animal)) +
  geom_line()

p2 <- ggplot(as.data.frame(res), aes(pred, prey)) +
  geom_point() +
  labs(y = "Prey", x = "Predators")

# To make this easier to understand - maybe
p2b <- ggplot(as.data.frame(res), aes(pred, prey, col = time)) +
  geom_point() +
  labs(y = "Prey", x = "Predators")

ggarrange(p1, p2b)
```





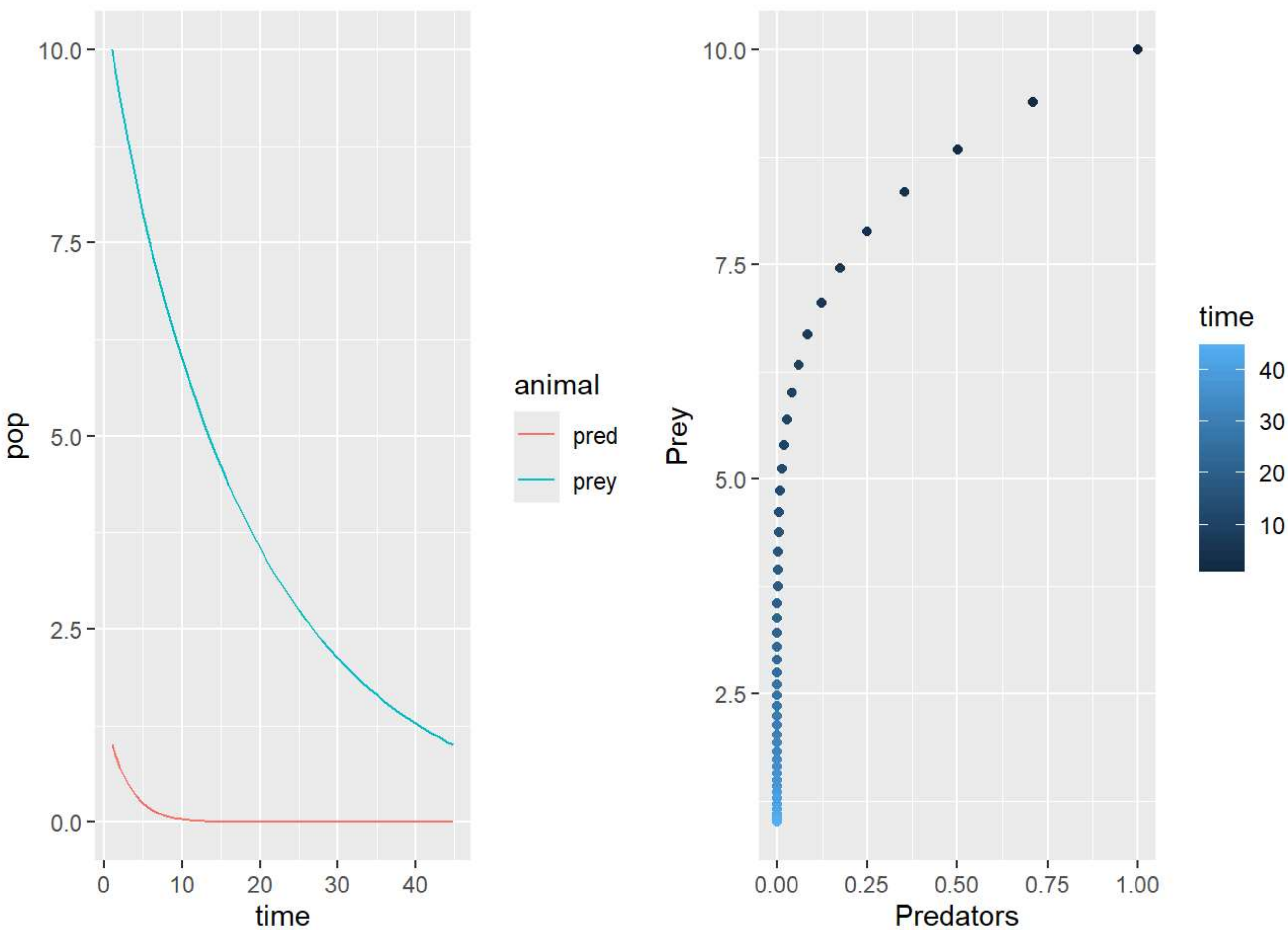
Second Model

```
# Change hunting parameters
pars <- c(rprey = 0.95, alpha = 0.01, eff = 0.6, pmort = 0.4, K = 2000, beta = 0.5, hunters = 2, 

# run the model
res <- ode(func = lotvmod_hunting, y = currpop, times = days, parms = pars)
```

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

In above message, R1 = 44.8981
```

```
# rearrange for easy plotting
resl <- as.data.frame(res) %>% pivot_longer(-time, names_to = "animal", values_to = "pop")
p1 <- ggplot(resl, aes(time, pop, col = animal)) +
  geom_line()

p2 <- ggplot(as.data.frame(res), aes(pred, prey)) +
  geom_point() +
```

```
  labs(y = "Prey", x = "Predators")

# To make this easier to understand - maybe
p2b <- ggplot(as.data.frame(res), aes(pred, prey, col = time)) +
  geom_point() +
  labs(y = "Prey", x = "Predators")

ggarrange(p1, p2b)
```



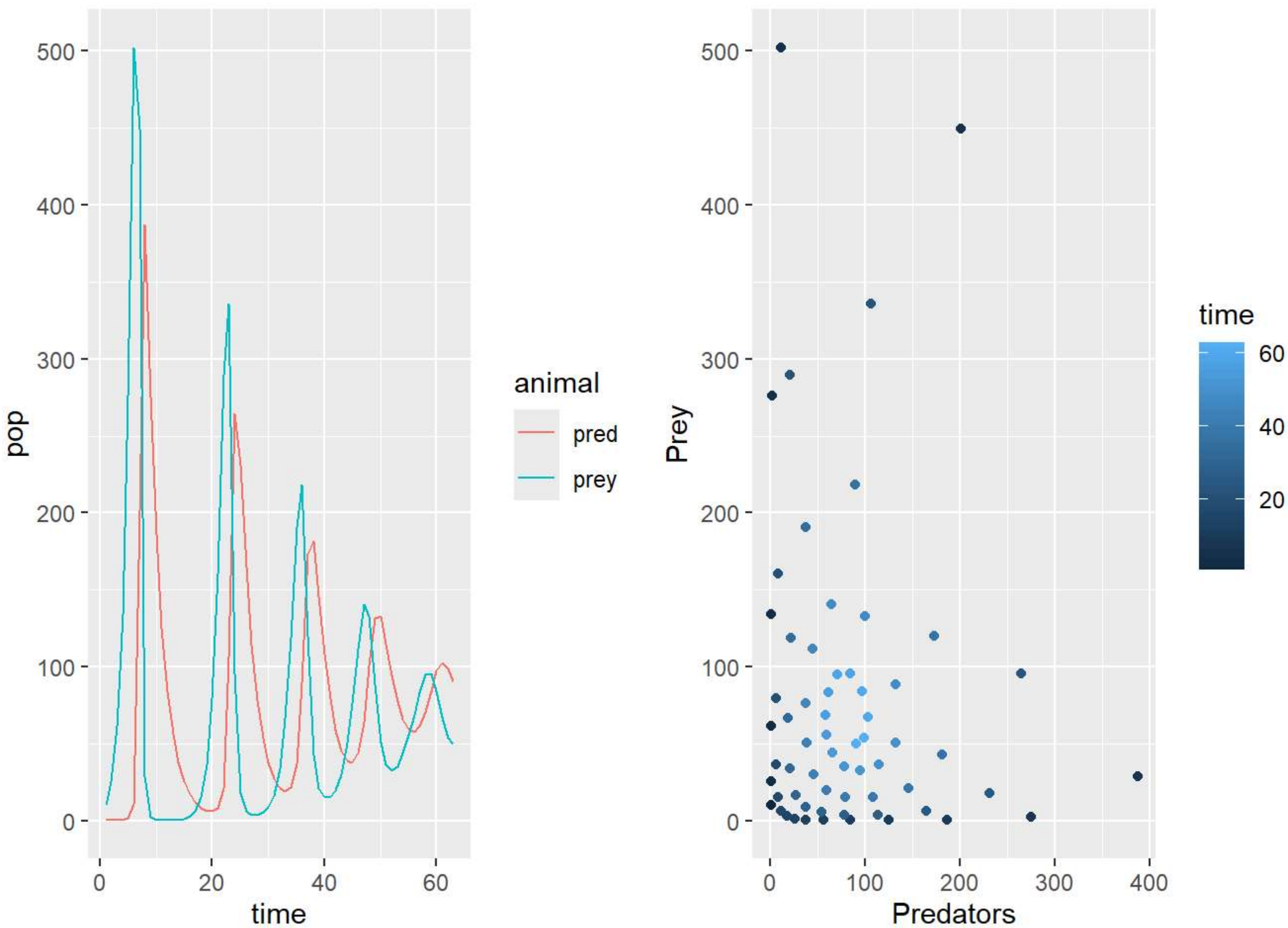
Third Model

```
# Change hunting parameters
pars <- c(rprey = 0.95, alpha = 0.01, eff = 0.6, pmort = 0.4, K = 2000, beta = 0.5, hunters = 0.2

# run the model
res <- ode(func = lotvmod_hunting, y = currpop, times = days, parms = pars)
```

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

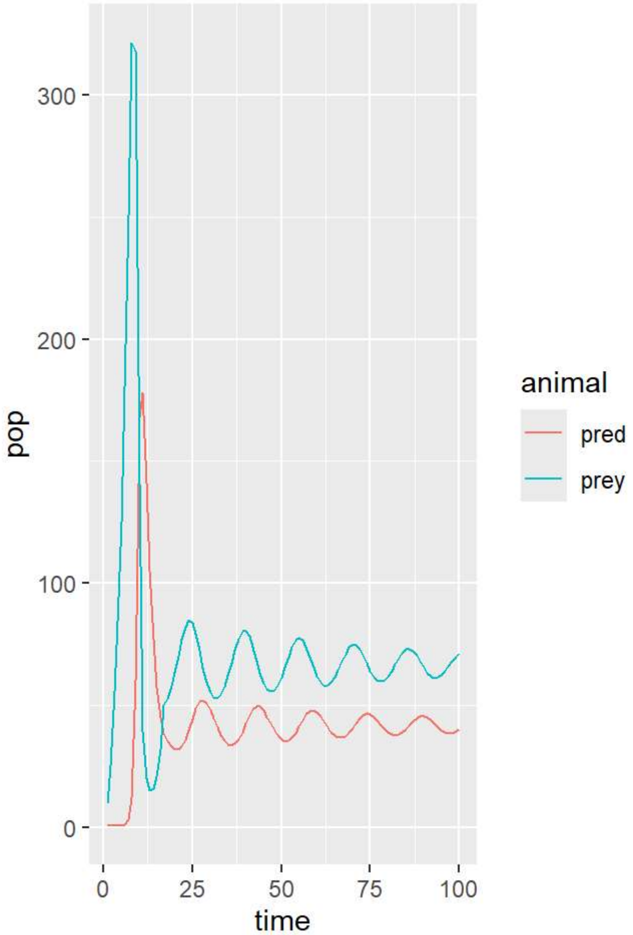
```
In above message, R1 = 62.9376
```

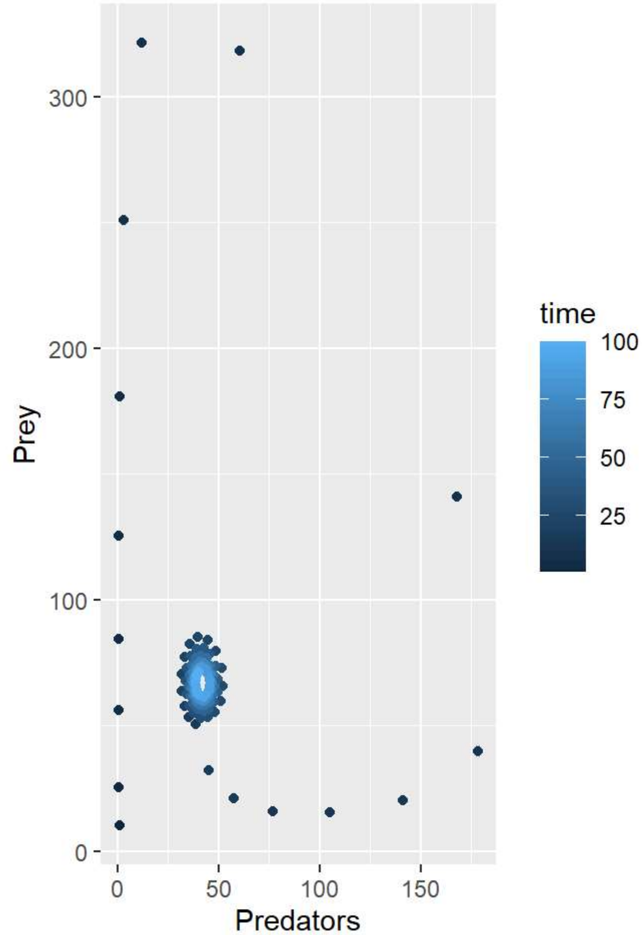
```
# rearrange for easy plotting
resl <- as.data.frame(res) %>% pivot_longer(-time, names_to = "animal", values_to = "pop")
p1 <- ggplot(resl, aes(time, pop, col = animal)) +
  geom_line()


p2 <- ggplot(as.data.frame(res), aes(pred, prey)) +
  geom_point() +
  labs(y = "Prey", x = "Predators")

# To make this easier to understand - maybe
p2b <- ggplot(as.data.frame(res), aes(pred, prey, col = time)) +
  geom_point() +
  labs(y = "Prey", x = "Predators")

ggarrange(p1, p2b)
```



Fourth Model

```
source(here("R/hunting_model.R"))

# note the use of with
# initial conditions
currpop <- c(prey = 10, pred = 1)

# time points to see results
days <- seq(from = 1, to = 100, by = 1)

# set parameters
pars <- c(rprey = 0.95, alpha = 0.01, eff = 0.6, pmort = 0.4, K = 2000, beta = 0.5, hunters = 1, 

# run the model
res <- ode(func = lotvmod_hunting, y = currpop, times = days, parms = pars)

# rearrange for easy plotting
resl <- as.data.frame(res) %>% pivot_longer(-time, names_to = "animal", values_to = "pop")
p1 <- ggplot(resl, aes(time, pop, col = animal)) +
  geom_line()


p2 <- ggplot(as.data.frame(res), aes(pred, prey)) +
  geom_point() +
  labs(y = "Prey", x = "Predators")

# To make this easier to understand - maybe
p2b <- ggplot(as.data.frame(res), aes(pred, prey, col = time)) +
  geom_point() +
  labs(y = "Prey", x = "Predators")

ggarrange(p1, p2b)
```





Conclusions

The first model in Part 2 best represents the predator prey model without hunting. This is the most stable relationship with new parameters, beta = 0.5, hunters = 1, prey_thresh = 2 because they best represent the original predator prey model. The fourth model may also be considered stable because predator and prey populations slowly reach a static value. This includes the same parameters as the first model but with prey_thresh = 50. Because of the high growth rate of prey, we are able to set a high threshold.