Hastings et al. (2018) Science

T.J. Clark

2020-08-12

Trying to recreate the figures

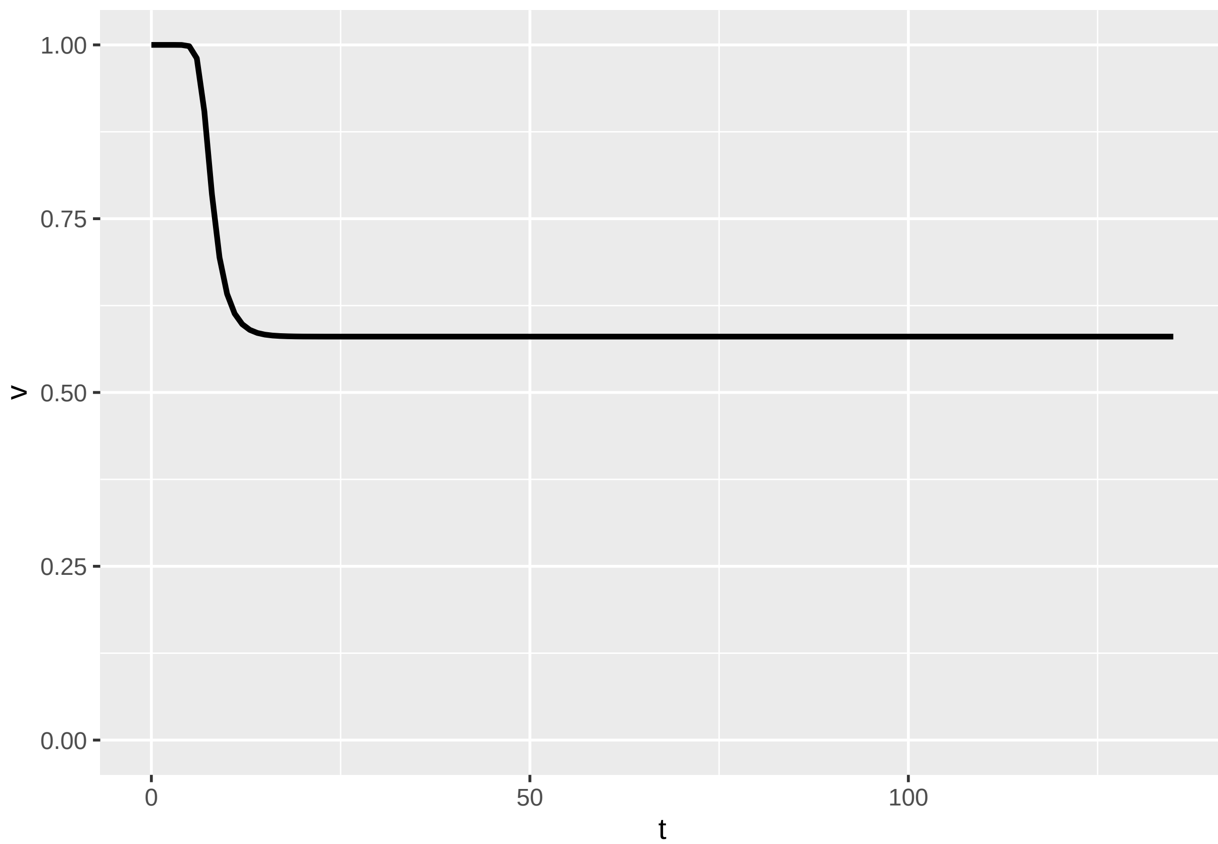
```
library(tidyverse)
library(deSolve)
library(nimble)
library(scatterplot3d)
```

Figure 1A-D - Competition Model showing Ghost Attractor

$$\frac{du}{dt} = u(1-u) - a_{12}u^n v$$
$$\frac{dv}{dt} = \gamma[v(1-v) - a_{21}u^n v]$$

Graph the ghost attractor in the competition model

```
comp <- function(Time, State, Pars){
  with(as.list(c(State, Pars)), {
    du <- u*(1-u) - a12*u^n*v
    dv <- gamma*(v*(1-v) - a21*u^n*v)
    return(list(c(du,dv)))
  })
}

# params
p <- c(a12 = 0.9, a21 = 1.1, gamma = 10, n=3) # Fig. 1A and C
p2 <- c(a12 = 0.9, a21 = 1.1, gamma = 10, n=1.8) # Fig. 1B and D
y <- c(u = .001, v = 1)
t <- seq(0,135,by=1)

out1 <- as_tibble(ode(y, t, comp, p)[,-1])
out2 <- as_tibble(ode(y, t, comp, p2)[,-1])

out1 %>%
  ggplot(aes(t, v)) +
  geom_line(lwd=1) +
  ylim(c(0,1))
```




```r
out2 %>%
  ggplot(aes(t, v)) +
  geom_line(lwd=1) +
  ylim(c(0,1))
```

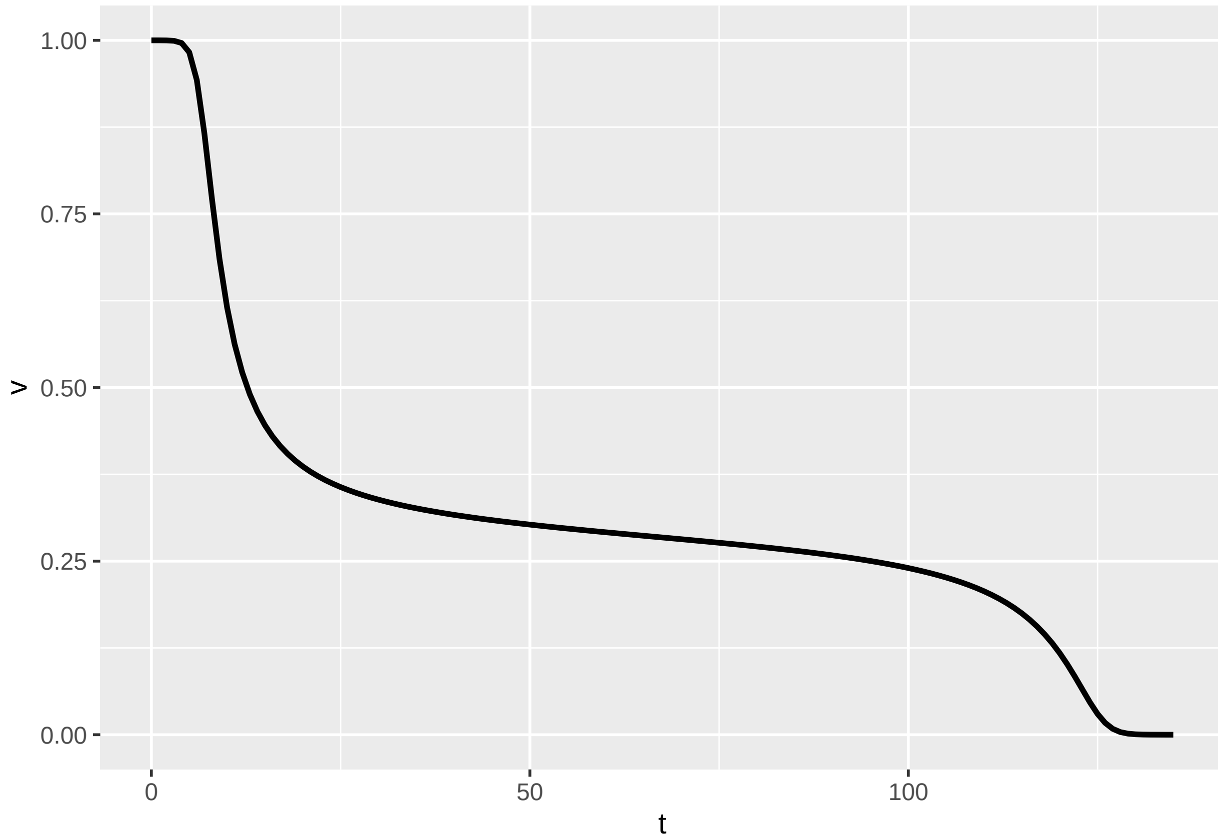

1.00
0.75
0.50
0.25
0.00
v
0
50
100
t

Figure 1E-F - 3 Species Chaotic Ghost Attractor

Originally from McCann & Yodzis (1994) American Naturalist

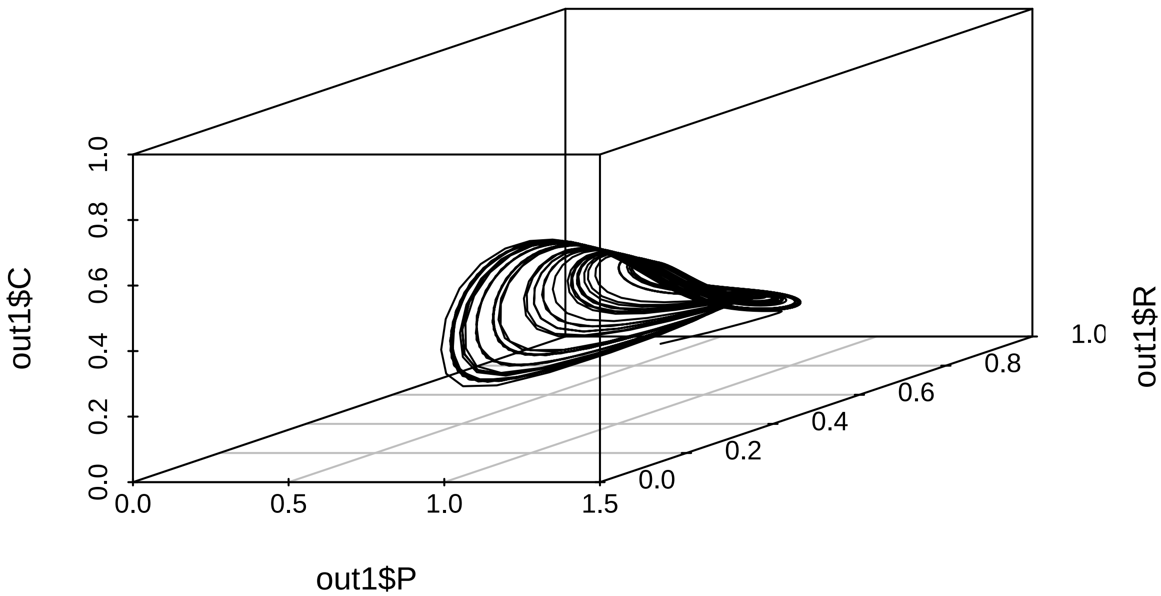
$$\frac{dR}{dt} = R[1 - (R/K)] - \frac{x_c y_c CR}{(R + R_0)}$$
$$\frac{dC}{dt} = x_c C([\frac{y_c R}{(R + R_0)}] - 1) - \frac{x_p y_p PC}{(C + C_0)}$$
$$\frac{dP}{dt} = x_p P([\frac{y_p C}{(C + C_0)}] - 1)$$

Graph

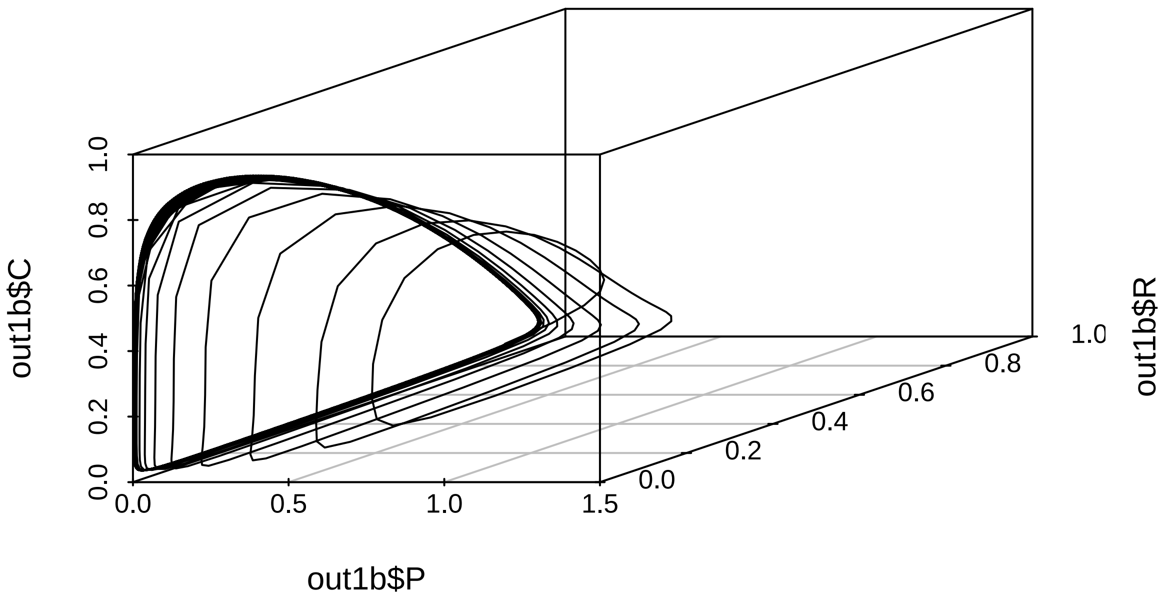
```
chaos <- function(Time, State, Pars){
  with(as.list(c(State, Pars)), {
    dR <- R*(1-(R/K)) - xc*yc*C*R/(R + R0)
    dC <- xc*C*((yc*R/(R + R0))-1) - xp*yp*P*C/(C + C0)
    dP <- xp*P*((yp*C/(C + C0))-1)
    return(list(c(dR,dC,dP)))
  })
}

# params
p <- c(K = 0.99, xc = 0.4, yc = 2.009, R0 = 0.16129, xp = 0.08, yp = 2.876, C0 = 0.5) # Fig. 1E and G
p2 <- c(K = 1.0, xc = 0.4, yc = 2.009, R0 = 0.16129, xp = 0.08, yp = 2.876, C0 = 0.5) # Fig. 1F and H
y <- c(R = .5, C = .2, P = 1)
y2 <- c(R = 0.5, C = 0.2, P = 0.5)
t <- seq(0,2000,by=1)

out1 <- as_tibble(ode(y, t, chaos, p)[,-1])
out1b <- as_tibble(ode(y2, t,chaos, p)[,-1])
out2 <- as_tibble(ode(y, t, chaos, p2)[,-1])
```

Demonstrate the state-space of K = 0.99. This shows both a chaotic attractor and a limit cycle with predator extinction. Bifurcation from the cycle to chaotic attractor is known as a “crisis” or a “blue sky catastrophe”.

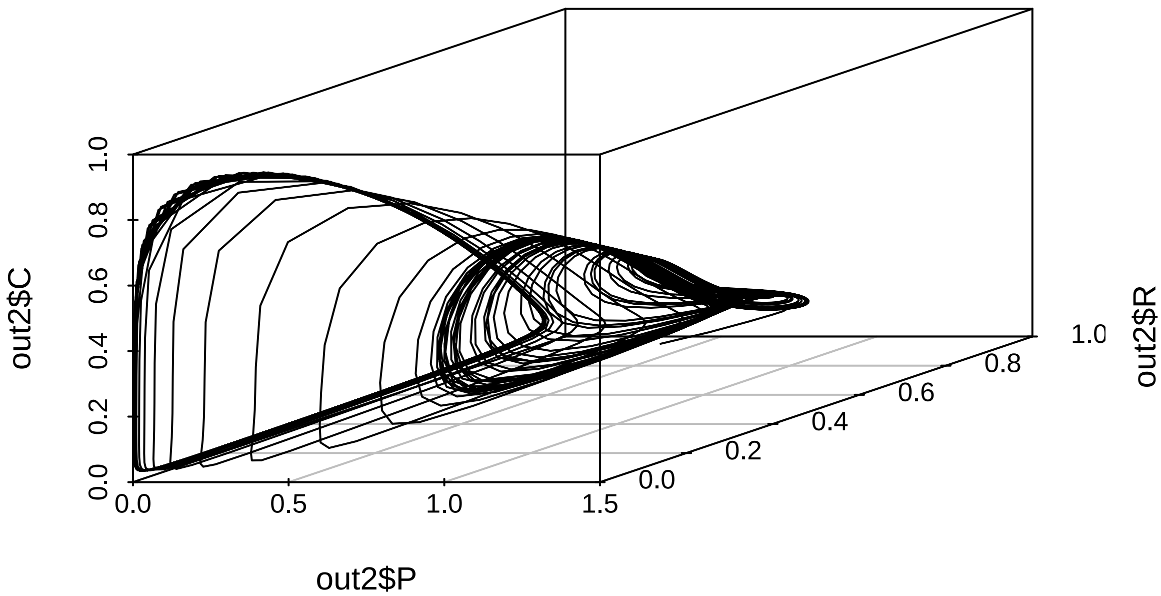
```
# state-space of chaotic/cyclic attractors

scatterplot3d(out1$P, out1$R, out1$C, type = "l",
              xlim = c(0,1.5), ylim = c(0,1), zlim = c(0,1))
```



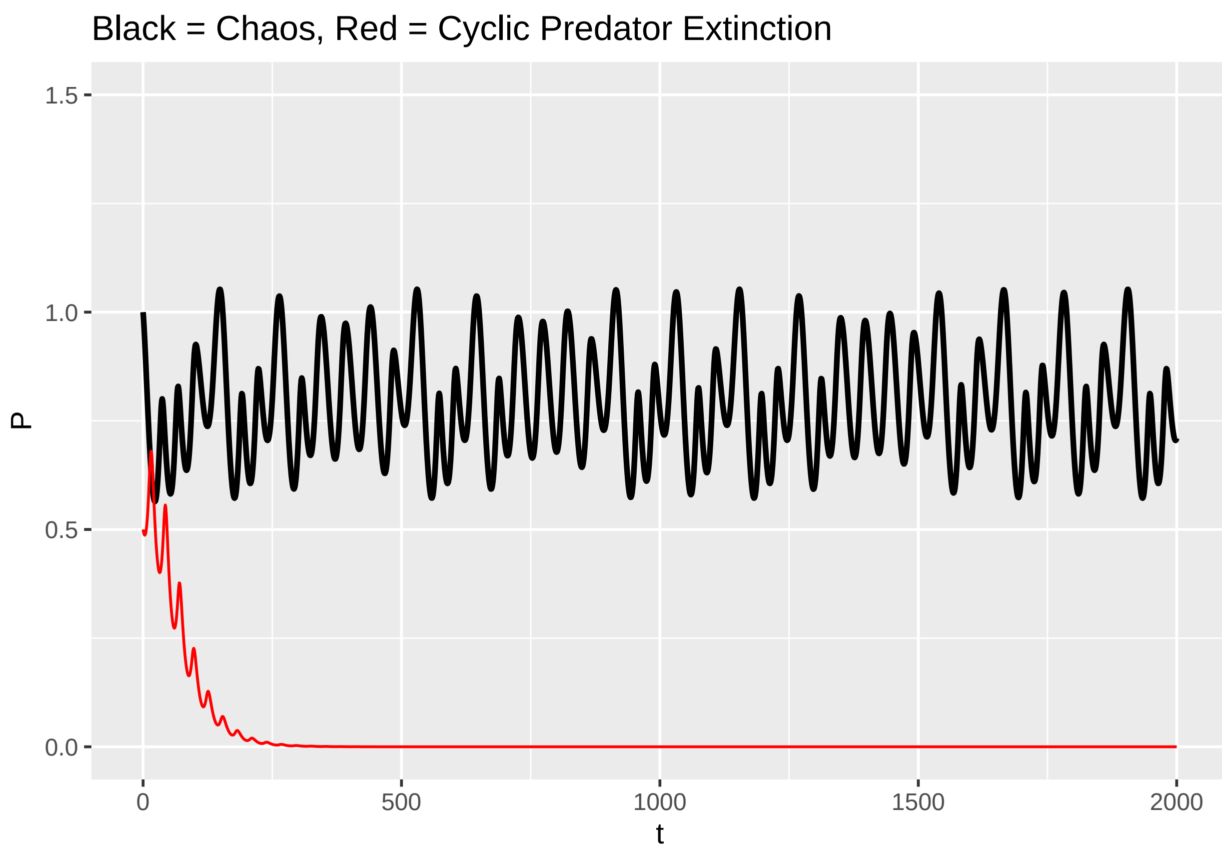

```
scatterplot3d(out1b$P, out1b$R, out1b$C, type = "l",
              xlim = c(0,1.5), ylim = c(0,1), zlim = c(0,1))
```




```r
scatterplot3d(out2$P, out2$R, out2$C, type = "l",
              xlim = c(0,1.5), ylim = c(0,1), zlim = c(0,1))
```




```
out1 %>%
  ggplot(aes(t, P)) +
  geom_line(lwd=1) +
  geom_line(aes(t,out1b$P), color = "red")+
  ylim(c(0,1.5)) +
  ggtitle("Black = Chaos, Red = Cyclic Predator Extinction")
```




```r
out2 %>%
  ggplot(aes(t, P)) +
  geom_line(lwd=1) +
  ylim(c(0,1.5)) +
  ggtitle("Long Chaotic Transient")
```

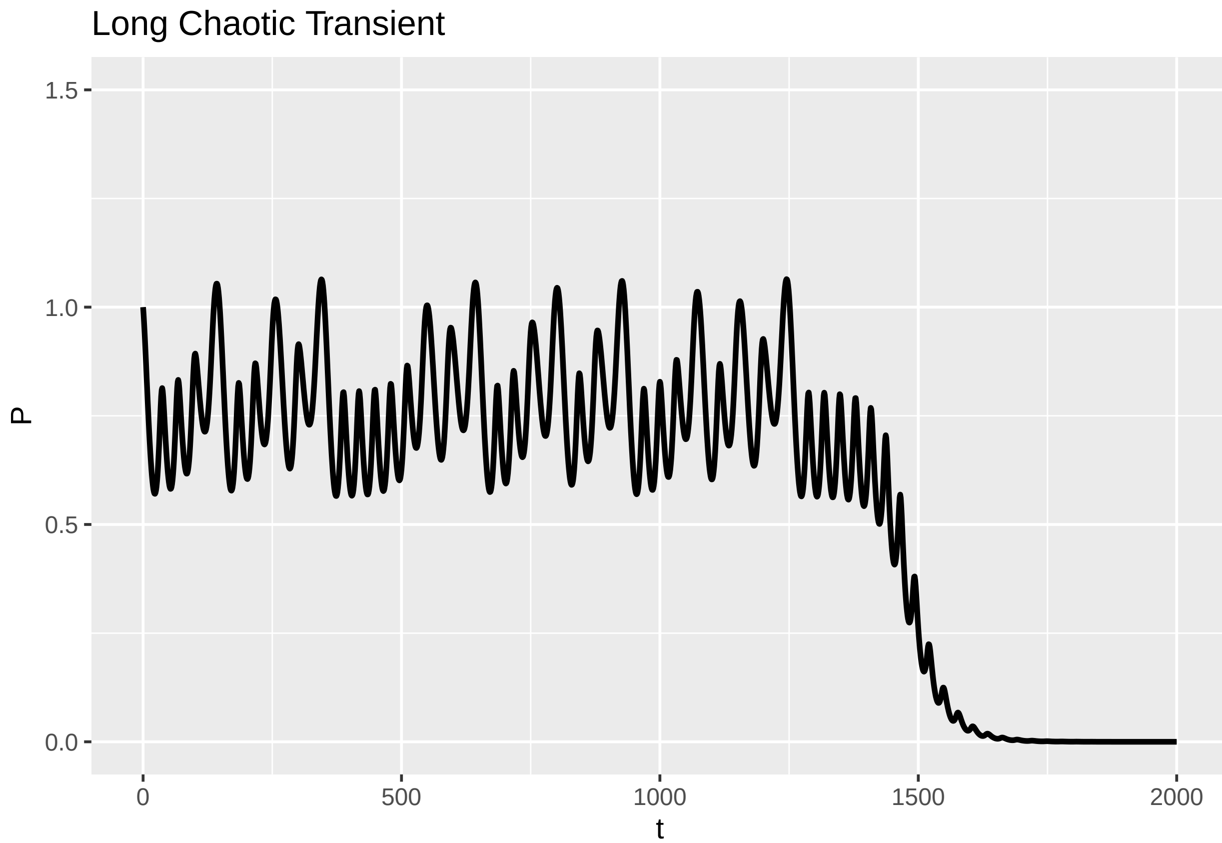

Long Chaotic Transient
P
1.5
1.0
0.5
0.0
0
500
1000
1500
2000
t

Figure 3 - Predator-Prey Transients due to Crawl-bys and Slow-fast Dynamics

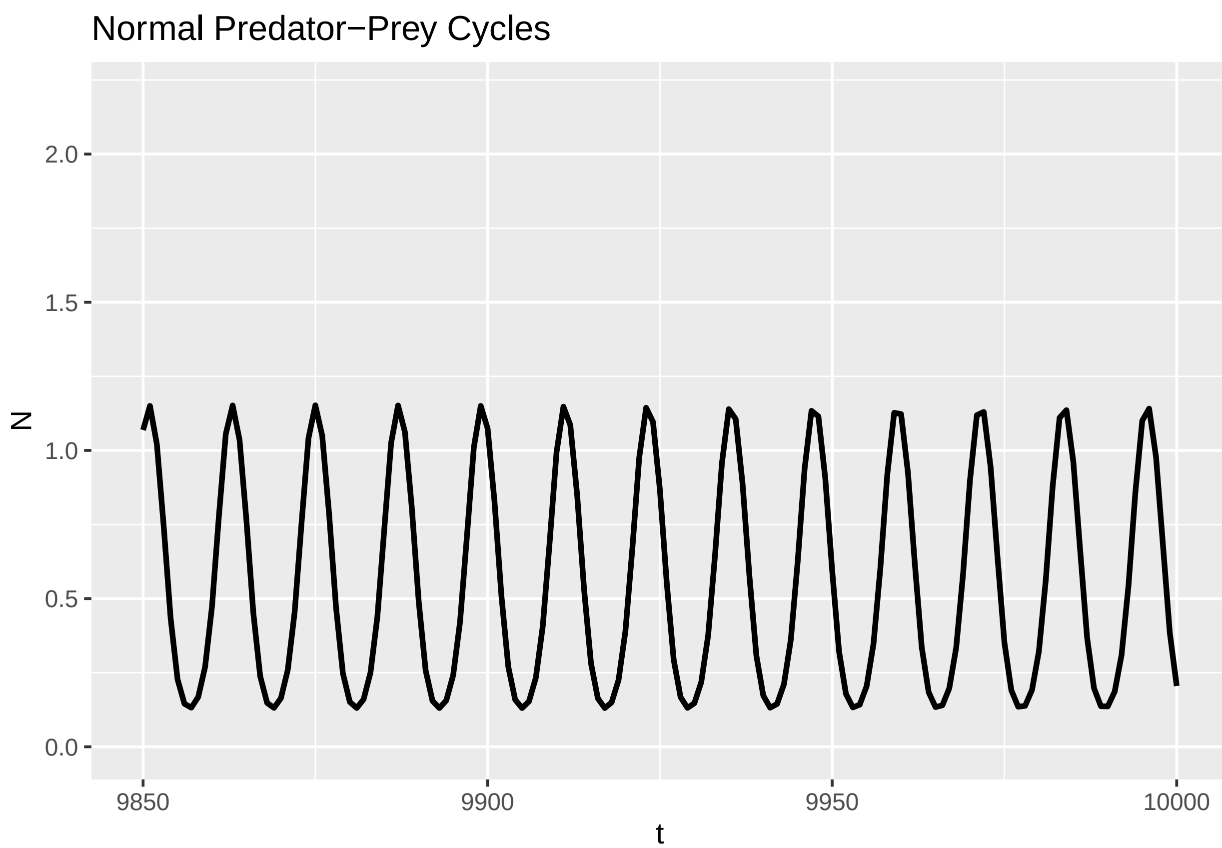
Epsilon is a scaling component that quanitifies order of magnitude for time scale of predator (P) and prey (N).

$$\frac{dN}{dt} = \alpha N[1 - (N/K)] - \frac{\gamma NP}{(N + H)}$$
$$\frac{dP}{dt} = \epsilon[(\frac{v\gamma NP}{(N + h)}) - mP]$$

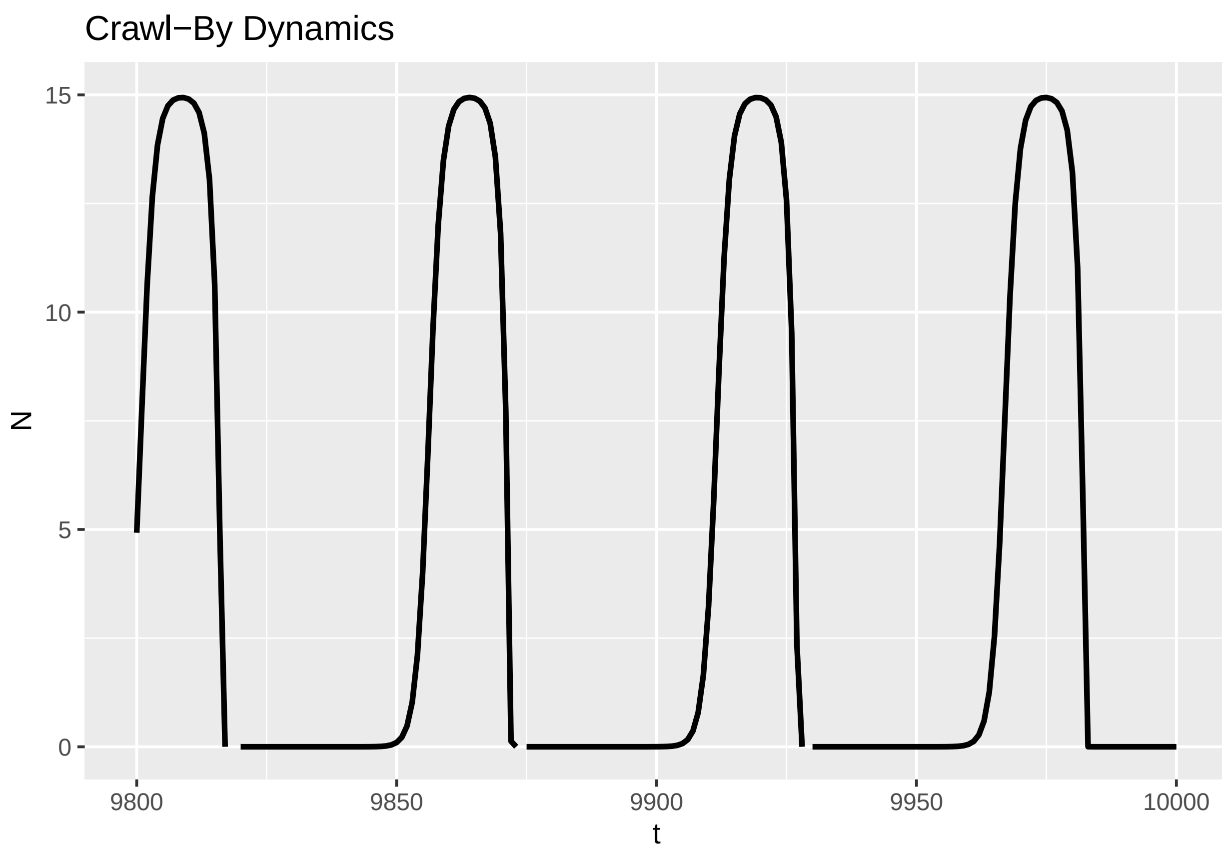
Graph

```
predprey <- function(Time, State, Pars){
  with(as.list(c(State, Pars)), {
    dN <- alpha*N*(1-(N/K)) - gamma*N*P/(N+h)
    dP <- epsilon*((v*gamma*N*P/(N+h)) - m*P)
    return(list(c(dN,dP)))
  })
}

# params
p <- c(gamma = 2.5, h = 1, v = 0.5, m = 0.4, alpha = 1.5, K = 2.2, epsilon = 1) # Fig. 3A and B
p2 <- c(gamma = 2.5, h = 1, v = 0.5, m = 0.4, alpha = 0.8, K = 15, epsilon = 1) # Fig. 3C and D
p3 <- c(gamma = 2.5, h = 1, v = 0.5, m = 0.4, alpha = 1.5, K = 2.2, epsilon = 0.01) # Fig. 3E and F
y <- c(N = 0.01, P = 0.01)
t <- seq(0,10000,by=1)

out1 <- as_tibble(ode(y, t, predprey, p)[,-1])
out2 <- as_tibble(ode(y, t, predprey, p2,
                      method = "ode45")[,-1])
out3 <- as_tibble(ode(y, t, predprey, p3)[,-1])
```

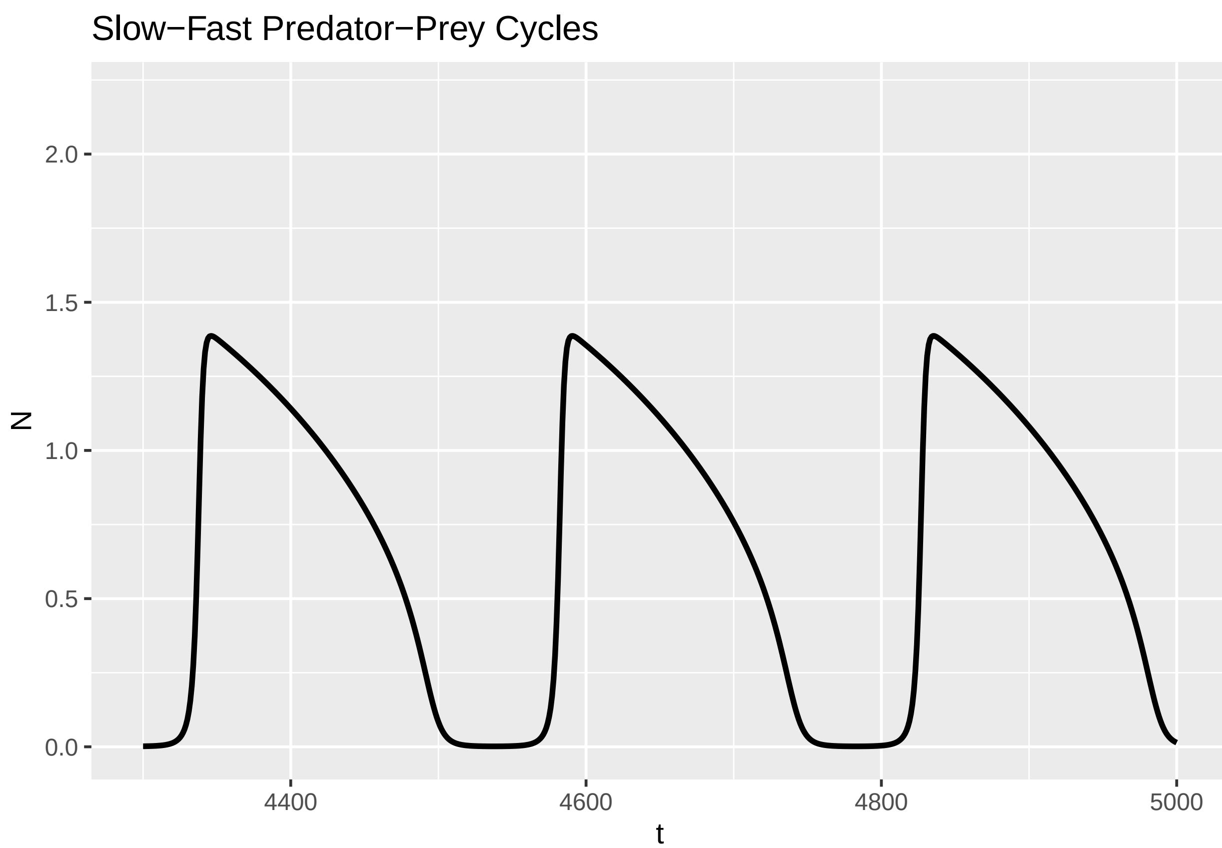
```
out1 %>%
  ggplot(aes(t, N)) +
  geom_line(lwd=1) +
  xlim(c(9850,10000)) +
  ylim(c(0,2.2)) +
  ggtitle("Normal Predator-Prey Cycles")
```




```r
out2 %>%
  ggplot(aes(t, N)) +
  geom_line(lwd=1) +
  xlim(c(9800,10000)) +
  ylim(c(0,15)) +
  ggtitle("Crawl-By Dynamics")
```




```r
out3 %>%
  ggplot(aes(t, N)) +
  geom_line(lwd=1) +
  xlim(c(4300,5000)) +
  ylim(c(0,2.2)) +
  ggtitle("Slow-Fast Predator-Prey Cycles")
```



Stochasticity Added to the Predator-Prey Model

Graph

```
# create artifical time series
t <- seq(0,200,by=1)
signal <- data.frame(t = t, alpha_rand = rnorm(length(t),1.25,0.25))
# create function to add in parameters
input <- approxfun(x = signal$t, y = signal$alpha_rand,
                   method = "constant", rule = 2)

predprey2 <- function(Time, State, Pars){
  with(as.list(c(State, Pars)), {

    # varying parameter by time
    alpha <- input(Time)

    dN <- alpha*N*(1-(N/K)) - gamma*N*P/(N+h)
    dP <- epsilon*((v*gamma*N*P/(N+h)) - m*P)
    return(list(c(dN,dP)))
  })
}

# params
p <- c(gamma = 2.5, h = 1, v = 0.5, m = 0.4, alpha = 1.5, K = 1.5, epsilon = 1)
```

```r
p2 <- c(gamma = 2.5, h = 1, v = 0.5, m = 0.4, K = 1.5, epsilon = 1)
y <- c(N = 0.01, P = 0.01)

out1 <- as_tibble(ode(y, t, predprey, p)[,-1])
out1b <- as_tibble(ode(y, t, predprey2, p2)[,-1])
```

```r
out1 %>%
  ggplot(aes(t, N)) +
  geom_line(lwd=1) +
  geom_line(aes(t,out1b$N), color = "red")+
  ylim(c(0,2.2)) +
  ggtitle("Black = Normal Predator-Prey Cycles, Red = Stochasticity-driven Cycles")
```