

Predator Prey Example

Jessica Burnett and Nathaniel Price

2019-01-29

1 Two-Species Model System

This example is based on the system described in Exploratory Data Analysis Using Fisher Information (Mayer et al. 2007; Frieden and Gatenby 2007, 229–31). The two species Lotka-Volterra differential equations for the system are

$$\frac{dx_1}{dt} = g_1 x_1 \left(1 - \frac{x_1}{k}\right) - \frac{l_{12} x_1 x_2}{1 + \beta x_1}$$
$$\frac{dx_2}{dt} = \frac{g_{21} x_1 x_2}{1 + \beta x_1} - m_2 x_2$$

The model parameters are $g_1 = m_2 = 1$, $l_{12} = g_{12} = 0.01$, $k = 625$, and $\beta = 0.005$. The initial conditions for the system were not provided in the original reference (Mayer et al. 2007). We found that $x_1 = 277.7815$ and $x_2 = 174.551$ provide reasonable results. The differential equations for the system are defined in R as shown below.

```
# Model parameters
parameters <- c(
  g1 = 1,
  m2 = 1,
  l12 = 0.01,
  g21 = 0.01,
  k = 625,
  B = 0.005
)

# Initial conditions
state <- c(
  x1 = 277.7815,
  x2 = 174.551
)

# System differential equations (eq. 7.17 and 7.18)
deq <- function(t, state, parameters) {
  with(as.list(c(state, parameters)), {
    dx1dt <- g1 * x1 * (1 - (x1 / k)) - (l12 * x1 * x2) / (1 + B * x1)
    dx2dt <- (g21 * x1 * x2) / (1 + B * x1) - m2 * x2
    list(c(dx1dt, dx2dt))
  })
}
```

We use the package `deSolve` to solve the system of differential equations. The original reference did not provide the time the system was observed (Mayer et al. 2007). We found that a complete cycle of the system corresponds to approximately 11.145 time units. The system of differential equations is solved and the system trajectory is plotted below.

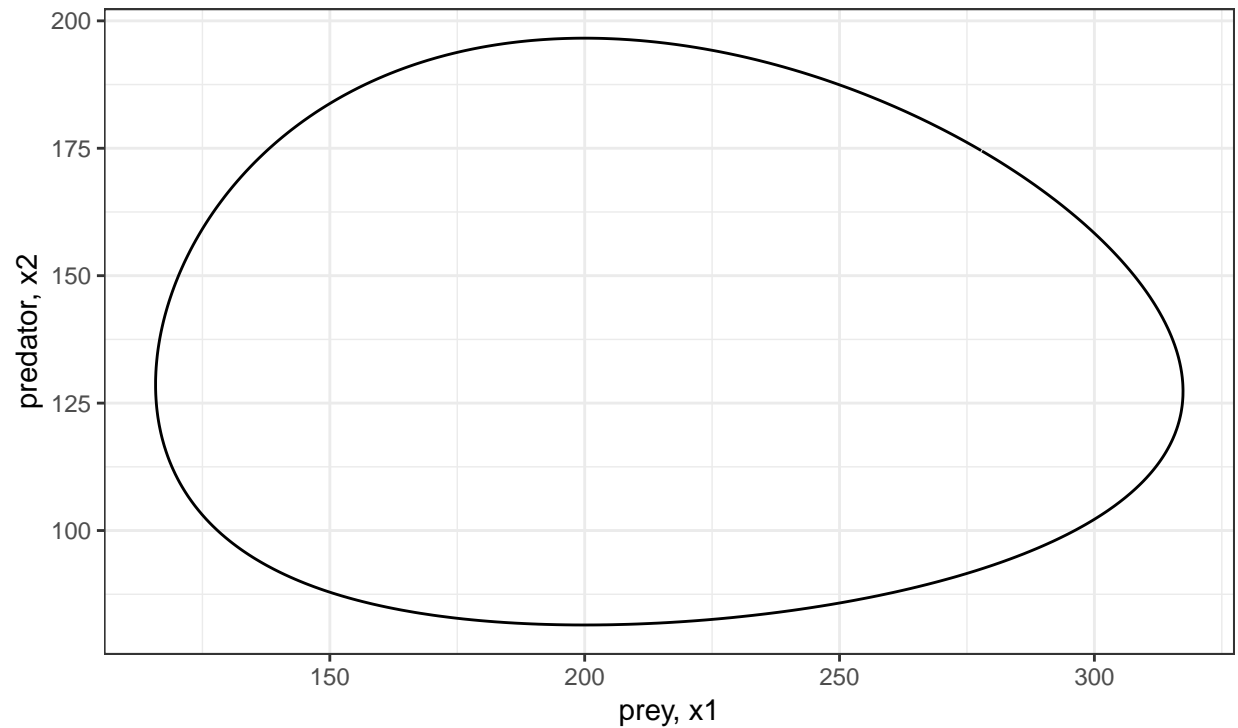


Figure 1: Phase space plot of system

```
# Vector of times
TT <- 11.145
times <- seq(0, TT, by = TT/1e3)

# Solve system differential equations
out <- ode(
  y = state,
  times
  = times,
  func = deq,
  parms = parameters,
  rtol = 1e-10,
  method = "ode45"
)

# Convert to data frame
sysSol <- data_frame(t = out[,1], x1 = out[,2], x2 = out[,3])

## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.
```

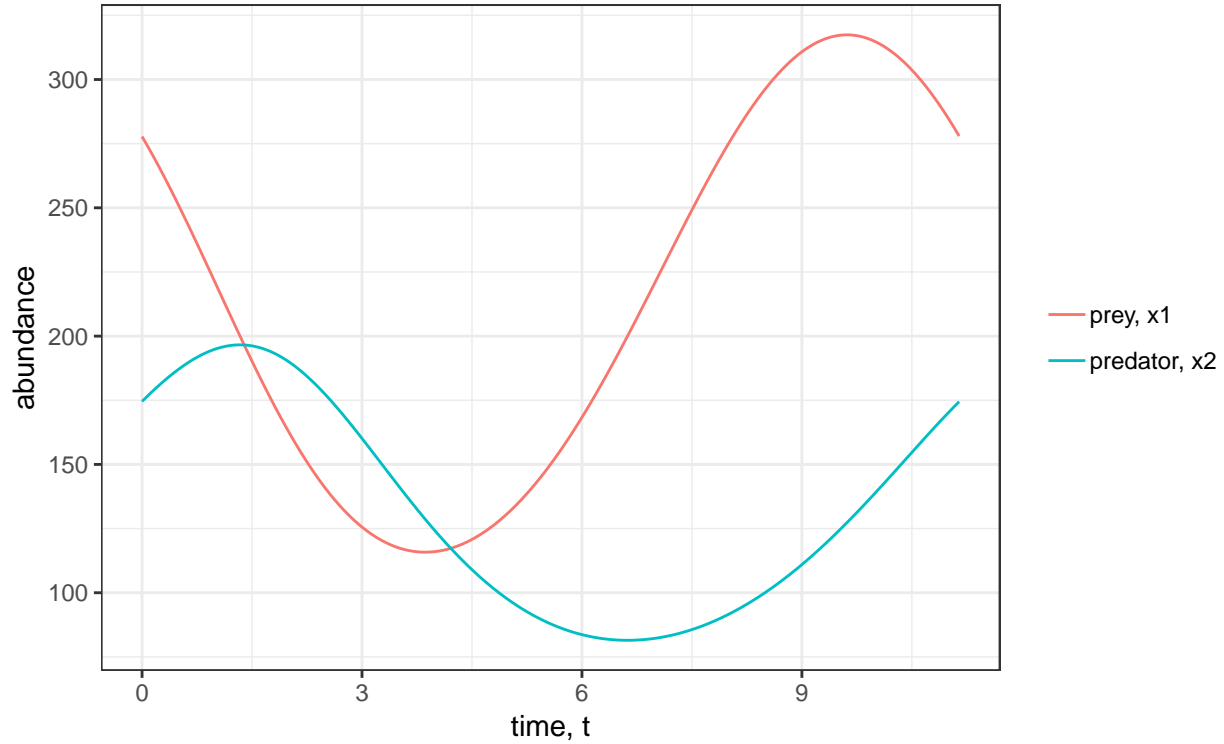


Figure 2: Change in predator and prey abundance over time

2 System State PDF & Dimensionality Reduction

The state of the system at any point in time, t , is defined in two dimensions by the number of prey, x_1 , and the number of predators x_2 . Let $\mathbf{x} = [x_1, x_2]$ represent the state of the system. The probability of *observing the system in a particular state* can be approximated based on a 2-dimensional histogram as shown below. Note that the system does not change at a constant rate (i.e., does not travel the path at a constant speed) and is therefore more likely to be observed in some states than others.

```
# Initial conditions including distance
```

```
state_high0 <- c(
  x1 = 277.7815,
  x2 = 174.551,
  dx1dt = -48.6,
  dx2dt = 28.3,
  d2x1dt2 = -23.3,
  d2x2dt2 = -10.5,
  s = 0
)
```

```
# Model parameters
```

```
parameters_high0 <- c(
  g1 = 1,
  m2 = 1,
  l12 = 0.01,
  g21 = 0.01,
  k = 625,

```

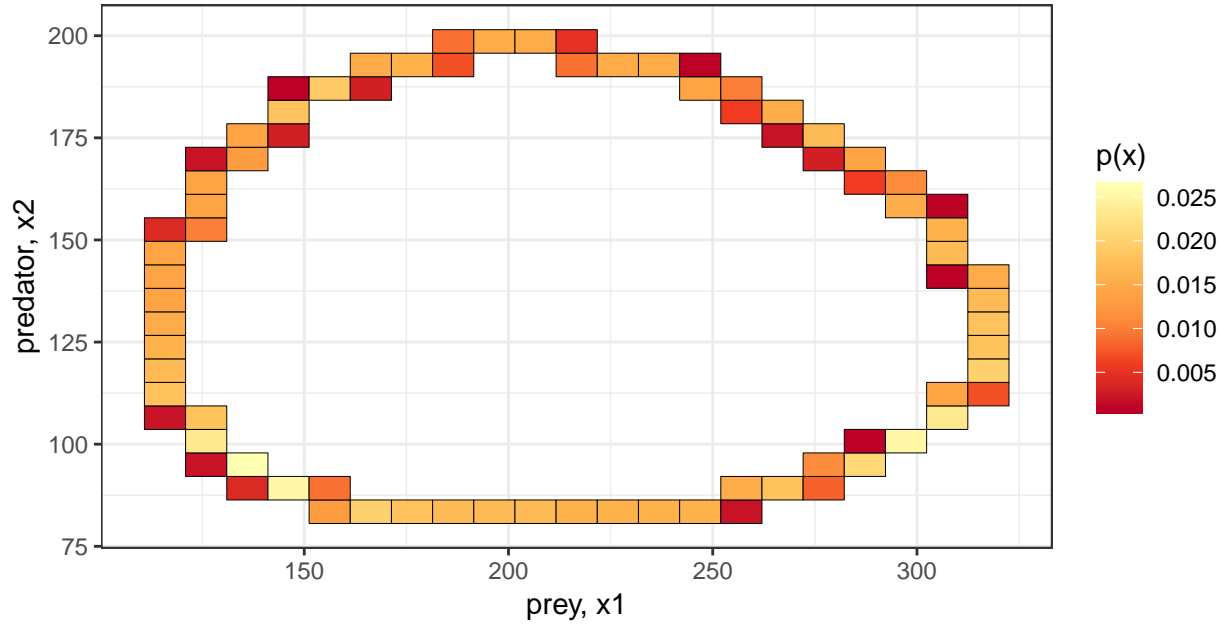


Figure 3: Normalized histogram showing probability of observing particular system states

```

B = 0.005
)

# System differential equations including higher order terms and distance
deq_high0 <- function(t, state, parameters) {
  with(as.list(c(state, parameters)), {

    # 1st derivatives
    dx1dt <-
      g1 * x1 * (1 - (x1 / k)) - (l12 * x1 * x2) / (1 + B * x1)

    dx2dt <- (g21 * x1 * x2) / (1 + B * x1) - m2 * x2

    # 2nd derivatives
    d2x1dt2 <-
      (B * l12 * x1 * x2 * dx1dt) / (B * x1 + 1) ^ 2 - (g1 * x1 * dx1dt) / k - (l12 *
      x1 * dx2dt) / (B * x1 + 1) - (l12 * x2 * dx1dt) / (B * x1 + 1) - g1 * (x1 /
      k - 1) * dx1dt

    d2x2dt2 <-
      (g21 * x1 * dx2dt) / (B * x1 + 1) - m2 * dx2dt + (g21 * x2 * dx1dt) / (B *
      x1 + 1) - (B * g21 * x1 * x2 * dx1dt) / (B * x1 + 1) ^ 2

    # 3rd derivatives
    d3x1dt3 <-
      (2 * B * l12 * x2 * dx1dt ^ 2) / (B * x1 + 1) ^ 2 - (2 * g1 * dx1dt ^ 2) /

```

```

k - (l12 * x1 * d2x2dt2) / (B * x1 + 1) - (l12 * x2 * d2x1dt2) / (B * x1 + 1) - (2 *
l12 * dx1dt * dx2dt) / (B * x1 + 1) - (g1 * x1 * d2x1dt2) / k - g1 * (x1 /
k - 1) * d2x1dt2 + (B * l12 * x1 * x2 * d2x1dt2) / (B * x1 + 1) ^ 2 - (2 *
B ^ 2 * l12 * x1 * x2 * dx1dt ^ 2) / (B * x1 + 1) ^ 3 + (2 * B * l12 * x1 *
dx1dt * dx2dt) / (B * x1 + 1) ^ 2

d3x2dt3 <-
(g21 * x1 * d2x2dt2) / (B * x1 + 1) - m2 * d2x2dt2 + (g21 * x2 * d2x1dt2) /
(B * x1 + 1) + (2 * g21 * dx1dt * dx2dt) / (B * x1 + 1) - (2 * B * g21 *
x2 * dx1dt ^ 2) / (B * x1 + 1) ^ 2 - (B * g21 * x1 * x2 * d2x1dt2) / (B *
x1 + 1) ^ 2 + (2 * B ^ 2 * g21 * x1 * x2 * dx1dt ^ 2) / (B * x1 + 1) ^ 3 - (2 *
B * g21 * x1 * dx1dt * dx2dt) / (B * x1 + 1) ^ 2

# Derivative of distance (i.e., tangential velocity)
dsdt <- sqrt(dx1dt ^ 2 + dx2dt ^ 2)

list(c(dx1dt, dx2dt, d2x1dt2, d2x2dt2, d3x1dt3, d3x2dt3, dsdt))
})
}

# Solve system differential equations (now with distance)
out_high0 <- ode(
  y = state_high0,
  times = times,
  func = deq_high0,
  parms = parameters_high0,
  rtol = 1e-10,
  method = "ode45"
)

# Convert to data frame
sysSol_high0 <-
  data_frame(t = out_high0[, 1],
             x1 = out_high0[, 2],
             x2 = out_high0[, 3],
             dx1dt = out_high0[,4],
             dx2dt = out_high0[,5],
             d2x1dt2 = out_high0[,6],
             d2x2dt2 = out_high0[,7],
             s = out_high0[, 8]) %>%
  # Calculate derivatives and pdf
  mutate(dsdt = sqrt(dx1dt^2 + dx2dt^2),
         d2sdt2 = (1/dsdt)*(dx1dt*d2x1dt2 + dx2dt*d2x2dt2),
         p = (1/TT)*(1/dsdt))

```

3 Fisher Information

The pdf of distance, $p(s)$, was derived above. There are several equivalent equations for calculating shift-invariant FI. Some may offer numerical advantages over others. The general form is (Mayer et al. 2007, eq. 7.3b)

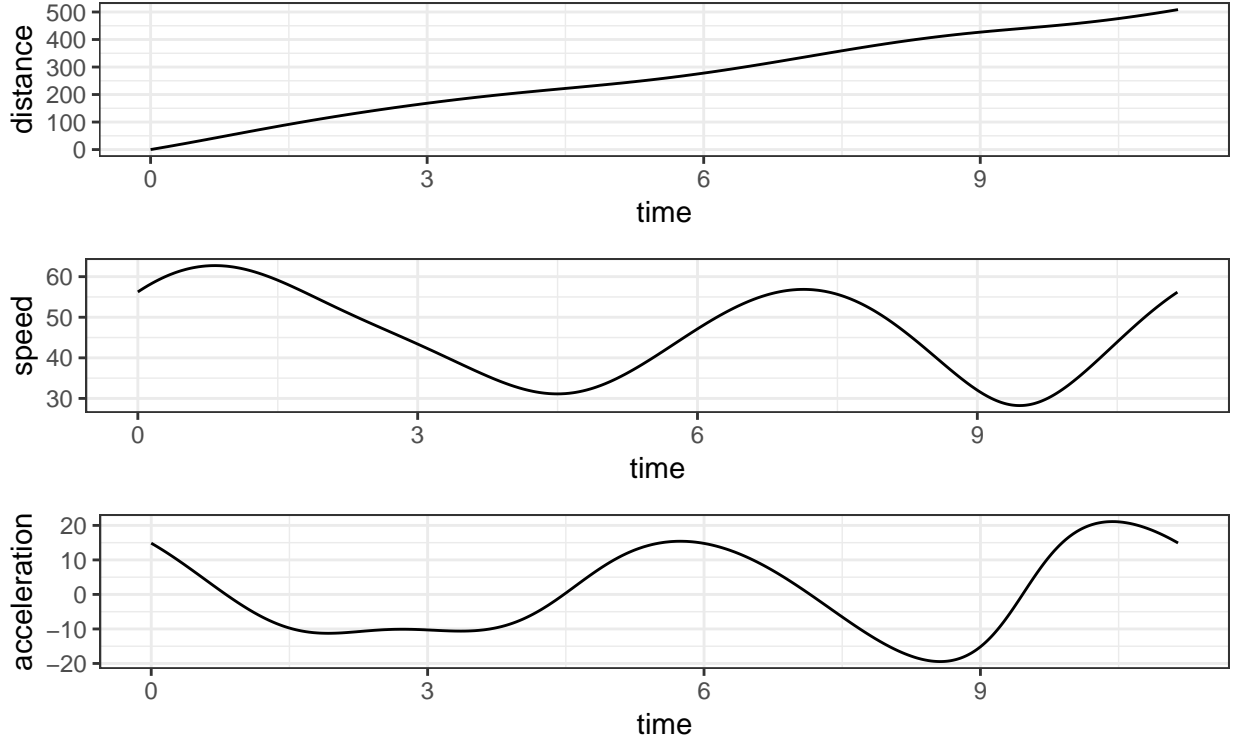


Figure 4: From top to bottom, distance traveled in phase space, speed tangential to system trajectory, acceleration tangential to system trajectory.

$$I = \int \frac{ds}{p(s)} \left[\frac{dp(s)}{ds} \right]^2$$

The amplitude form is (Mayer et al. 2007, eq. 7.3c)

$$I = 4 \int ds \left[\frac{dq(s)}{ds} \right]^2$$

A form specific to the pdf of distance traveled is derived as (Mayer et al. 2007, eq. 7.12)

$$I = \frac{1}{T} \int_0^T dt \left[\frac{s'^2}{s'^4} \right]^2$$

where integration is performed in the time domain. While all these formulations are equivalent, the last is most readily calculated when the differential equations for the system are known. Below all equations are shown to be equivalent. The FI for the distribution of distance is calculated to be approximately 5.4×10^{-5} which is consistent with published results (Mayer et al. 2007, 232).

```
# Equation 7.3b
p <- sysSol_high0$p
s <- sysSol_high0$s
dp <- lead(p)-p
ds <- lead(s)-s
dpds <- dp/ds
ind <- 1:(length(s)-1)
FI_7.3b <- trapz(s[ind], (1/p[ind])*dpds[ind]^2)
```

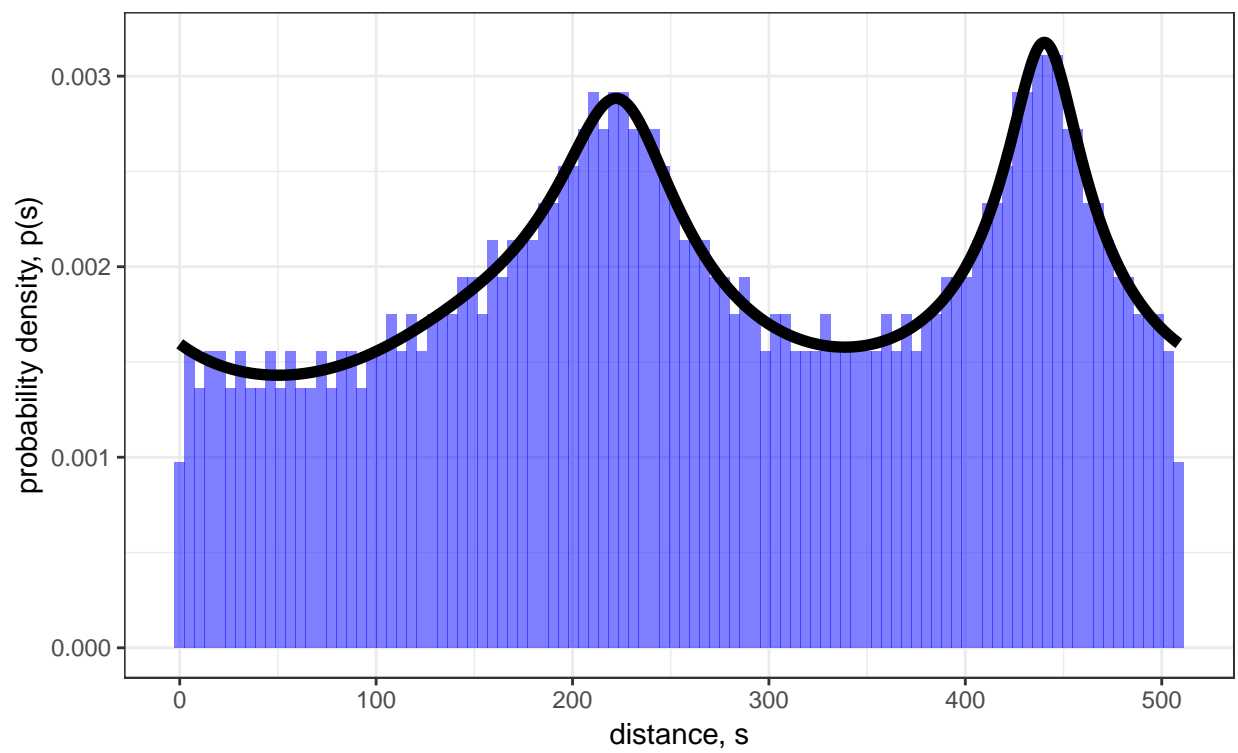


Figure 5: Comparison of normalized histogram of distance traveled (blue) to the derived probability density function (black)

```

# Equation 7.3c
q <- sqrt(sysSol_high0$p)
s <- sysSol_high0$s
dq <- lead(q)-q
ds <- lead(s)-s
dqds <- dq/ds
ind <- 1:(length(s)-1)
FI_7.3c <- 4*trapz(s[ind], dqds[ind]^2)

# Equation 7.12
t <- sysSol_high0$t
dsdt <- sysSol_high0$dsdt
d2sdt2 <- sysSol_high0$d2sdt2
ind <- 1:(length(s)-1)
FI_7.12 <- (1/TT)*trapz(t[ind], d2sdt2^2 / dsdt^4)

# Results
FI_7.3b

## [1] 5.370485e-05
FI_7.3c

## [1] 5.371751e-05
FI_7.12

## [1] 5.326461e-05

```

4 Varying carrying capacity for the prey

Here we replicate the results of varying the prey carrying capacity, k (Mayer et al. 2007).

5 Regime shift - Habitat destruction

Simulate a regime shift by abruptly decreasing the carrying capacity for the prey. The change in carrying capacity for the prey over time was modeled as

$$k(t) = k_1 - 0.5(k_1 - k_2)(\tanh(\alpha(t - t^*)) + 1)$$

where k_1 is the initial carrying capacity, k_2 is the final carrying capacity, t^* is the time of the regime shift, and α is a parameter that controls how abruptly the carrying capacity changes. The rate of change of the carrying capacity is given by

$$k'(t) = 0.5\alpha(k_1 - k_2)(\tanh(\alpha(t - t^*))^2 - 1)$$

```

# Time of regime shift
tregime = 200

# Carrying capacity for regime 1 and regime 2
k1 <- 800

```

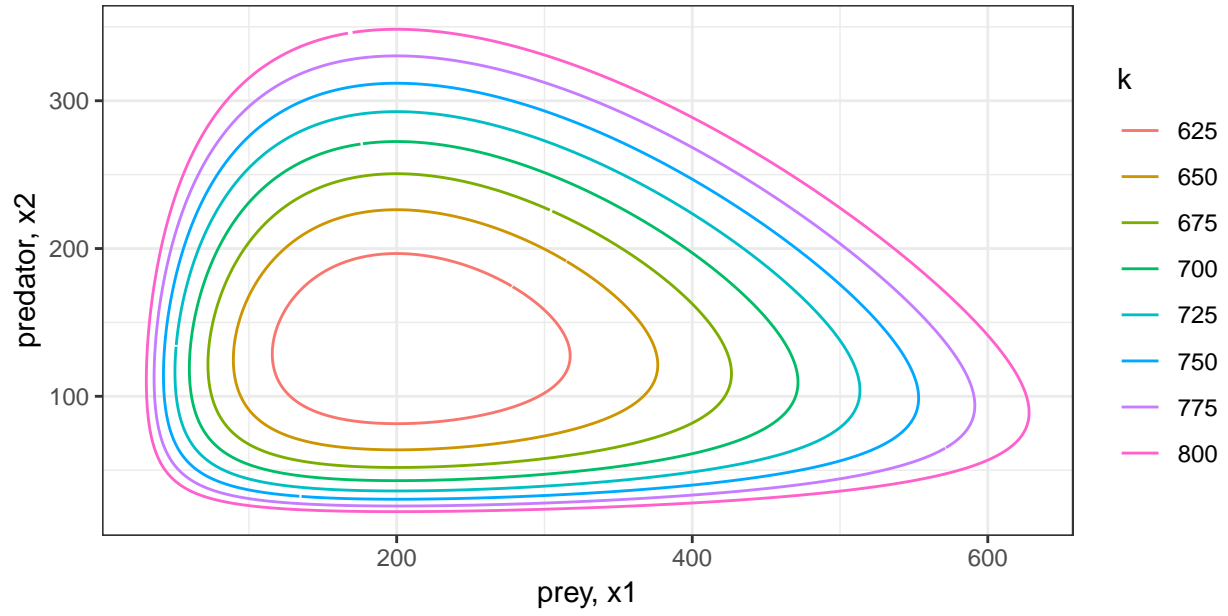



Figure 6: Phase space plot of system trajectories for different values of k

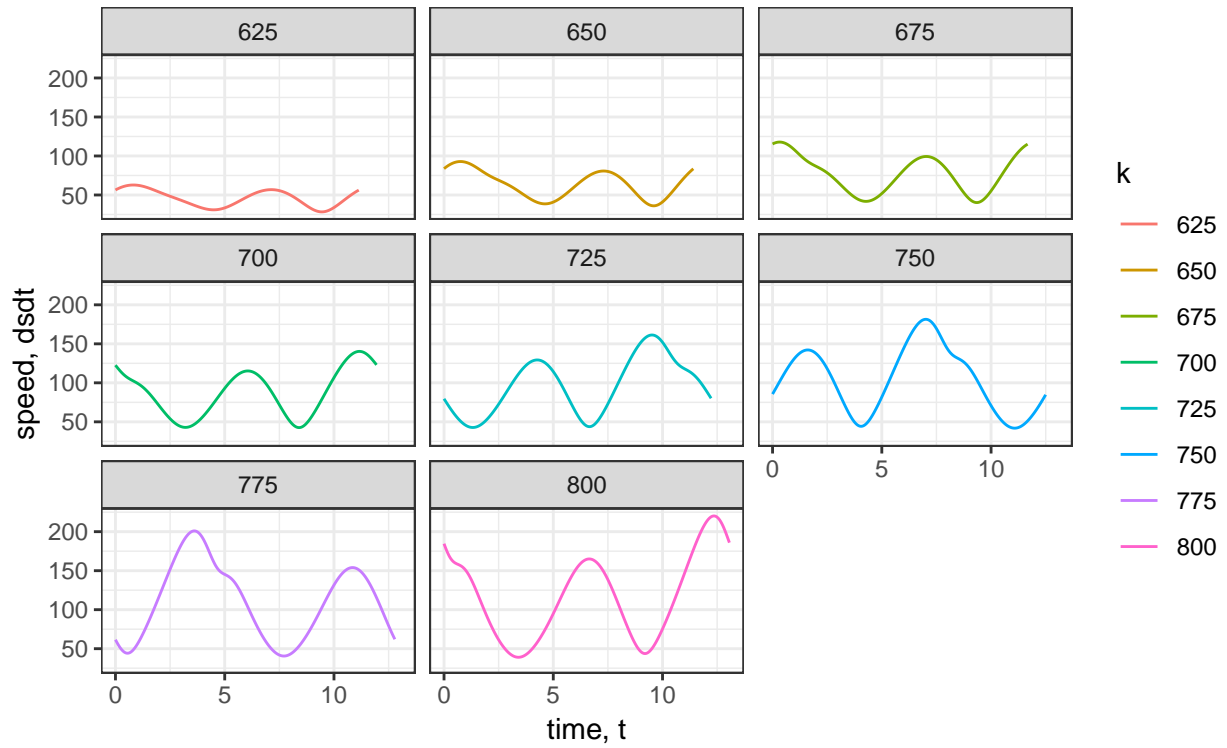


Figure 7: Plot of speed for different values of k

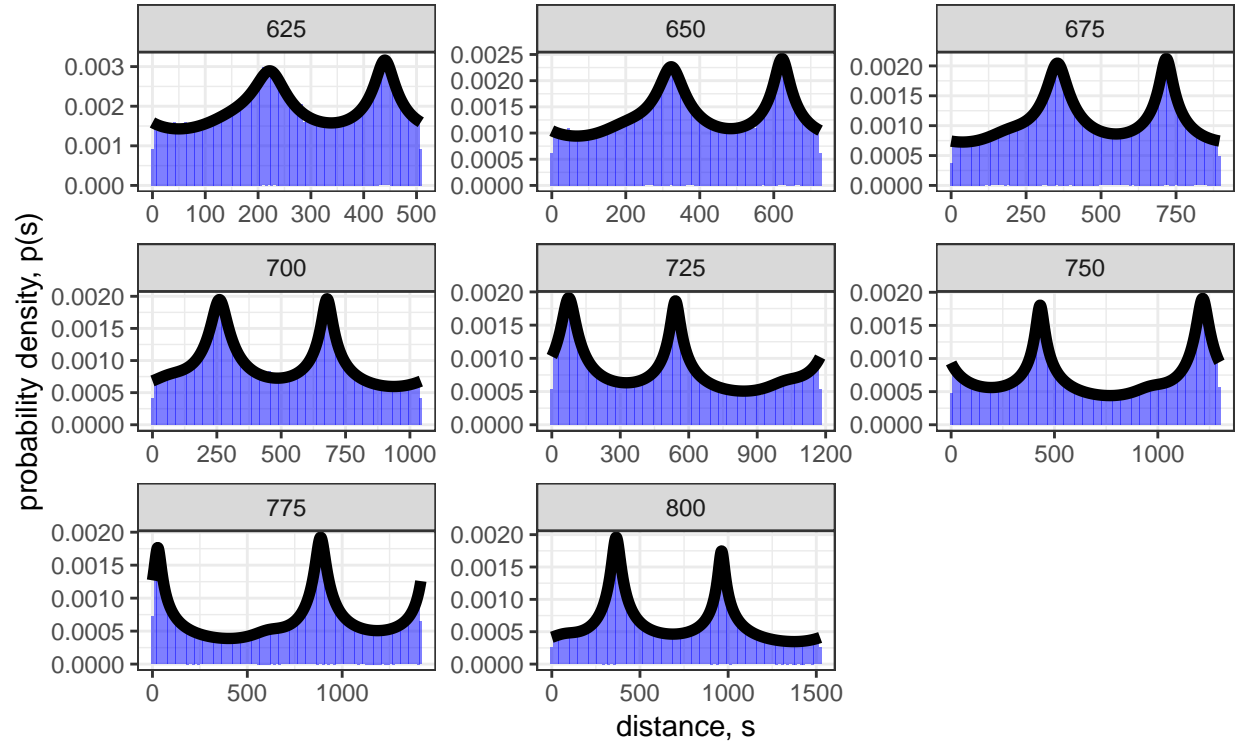


Figure 8: Plot distance pdf's for different values of k

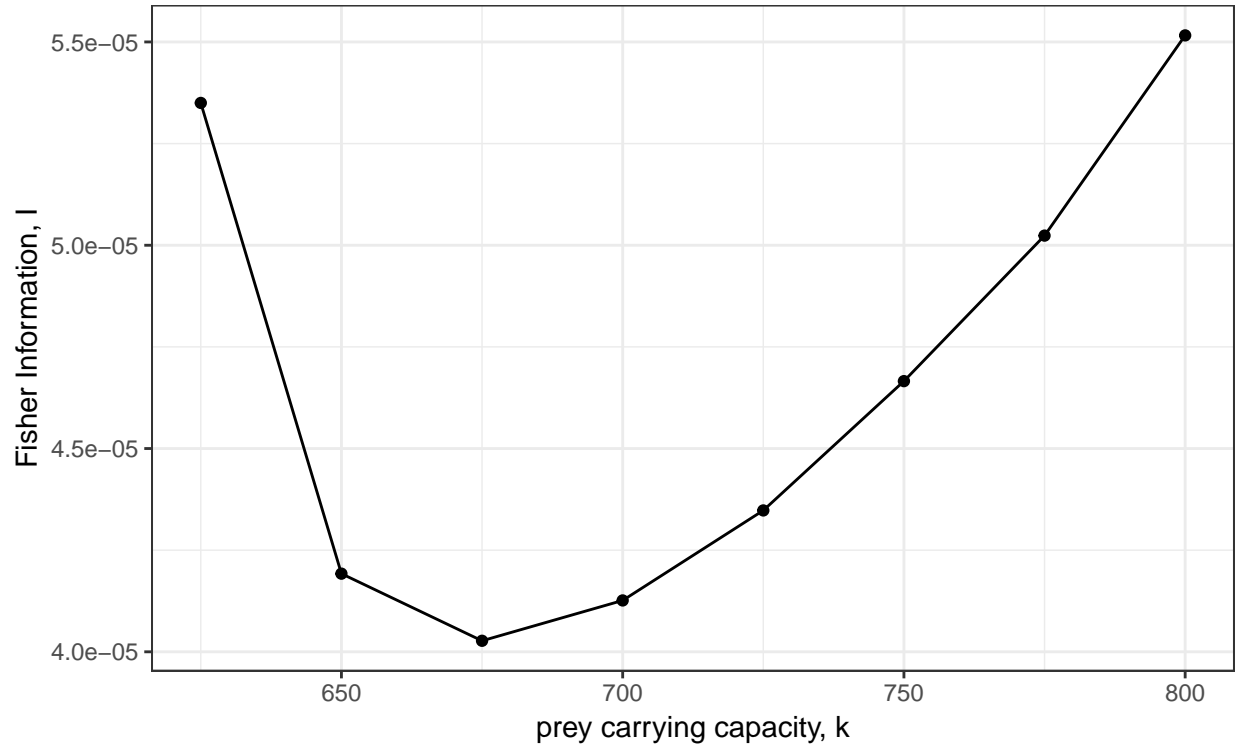


Figure 9: Plot of Fisher information for different values of k

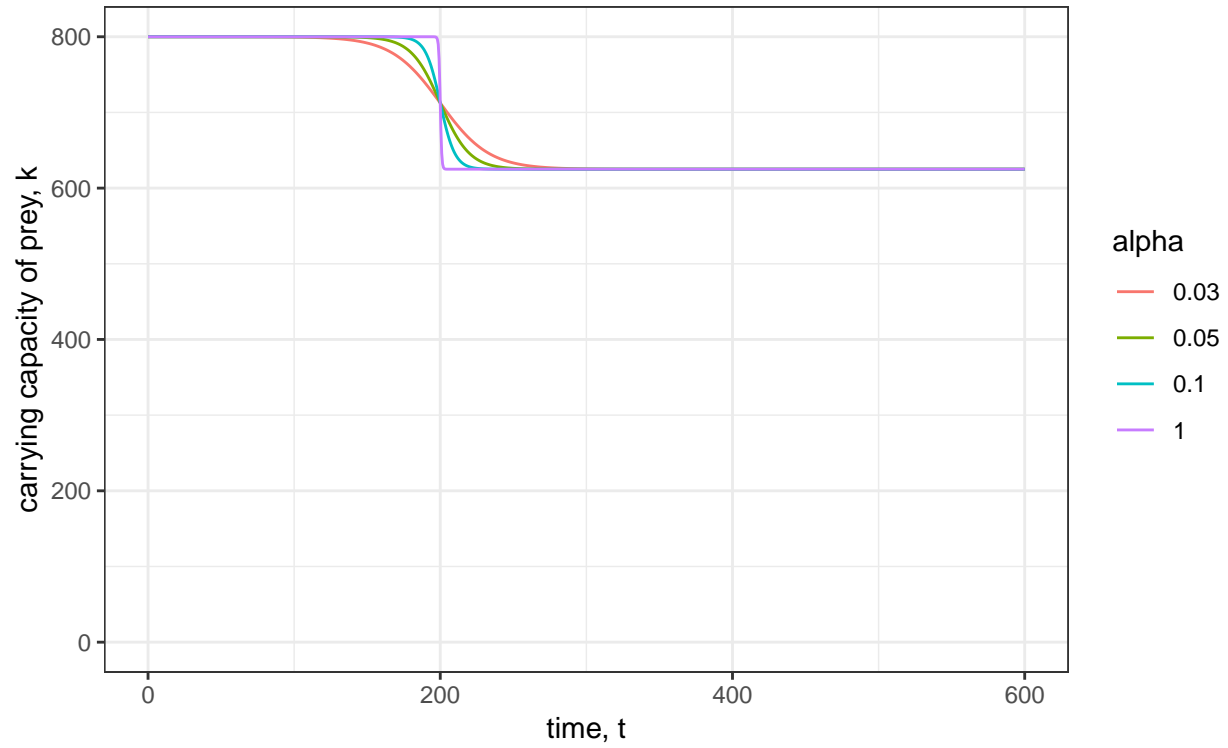


Figure 10: Change in carrying capacity of prey over time for different values of α

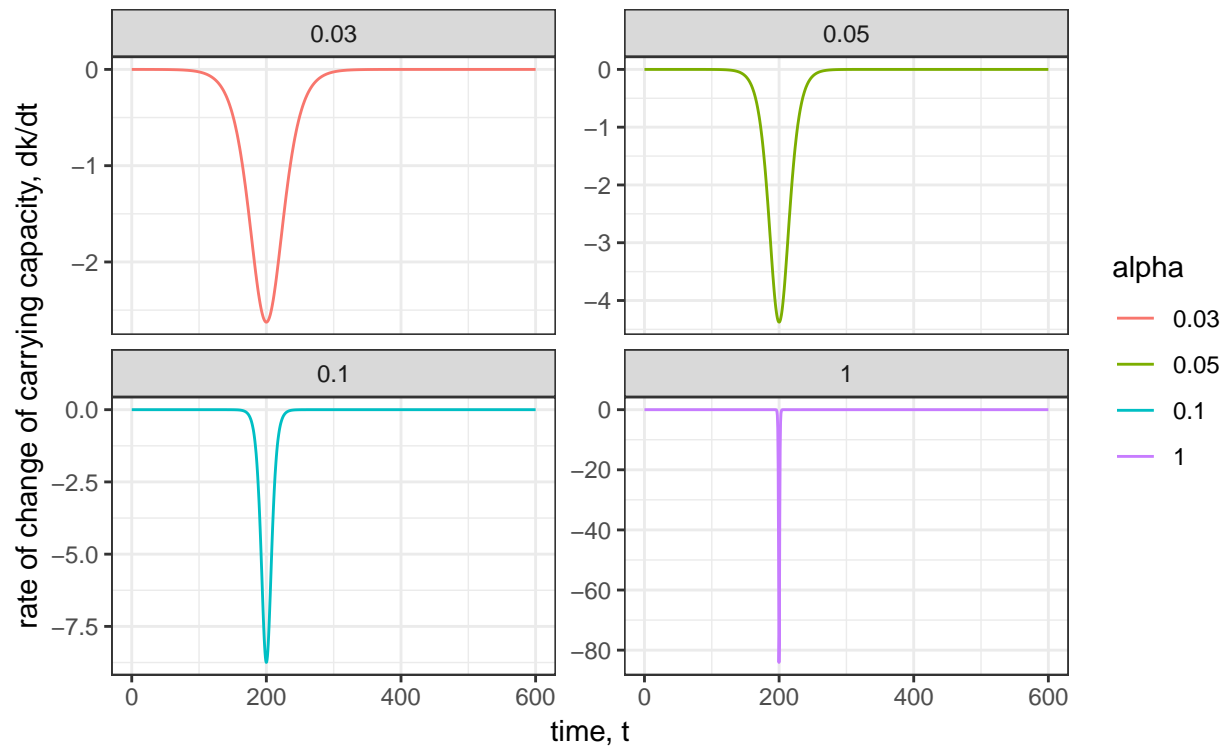


Figure 11: Rate of change of carrying capacity of prey over time for different values of α

```

k2 <- 625

# Regime change rate
alpha <- 0.05

# Vector of times
TT <- 600
times <- seq(0, TT, by = TT / 2e3)

# Model parameters
parameters <- c(
  g1 = 1,
  m2 = 1,
  l12 = 0.01,
  g21 = 0.01,
  B = 0.005
)

# Initial conditions
state <- c(
  x1 = 61.9,
  x2 = 35.3,
  s = 0,
  k = k1
)

# System differential equations including carrying capacity
deq_rs <- function(t, state, parameters) {
  with(as.list(c(state, parameters)), {

    # Predator and prey rates of change
    dx1dt <- g1 * x1 * (1 - (x1 / k)) - (l12 * x1 * x2) / (1 + B * x1)
    dx2dt <- (g21 * x1 * x2) / (1 + B * x1) - m2 * x2

    # Rate of change of distance
    dsdt <- sqrt(dx1dt ^ 2 + dx2dt ^ 2)

    # Rate of change of carrying capacity
    dkdt <- 0.5*alpha*(k1 - k2)*(tanh(alpha*(t - tregime))^2 - 1)

    list(c(dx1dt, dx2dt, dsdt, dkdt))
  })
}

# Solve system differential equations (now with carrying capacity)
out_rs <- ode(
  y = state,
  times = times,
  func = deq_rs,
  parms = parameters,
  rtol = 1e-10,
  method = "ode45"
)

```

```

)

# Convert to data frame
sysSol <-
  data_frame(
    t = out_rs[, 1],
    x1 = out_rs[, 2],
    x2 = out_rs[, 3],
    s = out_rs[, 4],
    k = out_rs[, 5]
  ) %>%
  mutate(dsdt = (lead(s) - s) / (lead(t) - t),
    p = (1 / maxT) * (1 / dsdt)) %>%
  filter(!is.na(dsdt))

# Distance over which to move the window (in units time)
winspaceFactor <- 1

# Size of the window (in units time)
# winsizeVector <- seq(13.1, 11.0, by = -0.25)
winsizeVector <- c(13.061, 11.135)

winResults <- NULL
for(i in 1:length(winsizeVector)){

  # Define winsize
  winsize <- winsizeVector[i]

  for(j in 1:length(winspaceFactor)) {
    winspace <- winspaceFactor[j]*winsize

    # Start and stop points for windows
    winStart <- seq(min(times), max(times) - winsize, by = winspace)
    winStop <- winStart + winsize

    # Number of windows
    nWin <- length(winStart)

    FI_7.3b <- numeric(length(nWin))
    avgSpeed <- numeric(length(nWin))

    for (win in 1:nWin) {
      df <-
        sysSol %>%
        filter(t > winStart[win],
          t <= winStop[win]) %>%
        mutate(TT = max(t) - min(t),
          p = (1 / TT) * (1 / dsdt))

      # Fisher information
      p <- df$p
      s <- df$s
      dp <- lead(p) - p
      ds <- lead(s) - s
    }
  }
}

```

```

dpds <- dp / ds
ind <- 1:(length(s) - 1)
FI_7.3b[win] <- trapz(s[ind], (1 / p[ind]) * dpds[ind] ^ 2)

# Average speed
avgSpeed[win] <- mean(df$dsdt)
}

tempResults <- data_frame(winStop = winStop,
FI = FI_7.3b,
avgSpeed = avgSpeed) %>%
mutate(winsize = winsize,
       winspace = winspace,
       winspaceFactor = winspaceFactor[j])

winResults <- rbind(winResults, tempResults)
}
}

winResults <-
  winResults %>%
  mutate(winsize = factor(winsize),
         winspace = factor(winspace))

```

References

- Frieden, B. Roy, and Robert Gatenby, eds. 2007. *Exploratory Data Analysis Using Fisher Information*. Springer. <http://www.springer.com/us/book/9781846285066>.
- Mayer, Audrey L., Christopher W. Pawlowski, Brian D. Fath, and Heriberto Cabezas. 2007. “Applications of Fisher Information to the Management of Sustainable Environmental Systems.” In *Exploratory Data Analysis Using Fisher Information*.

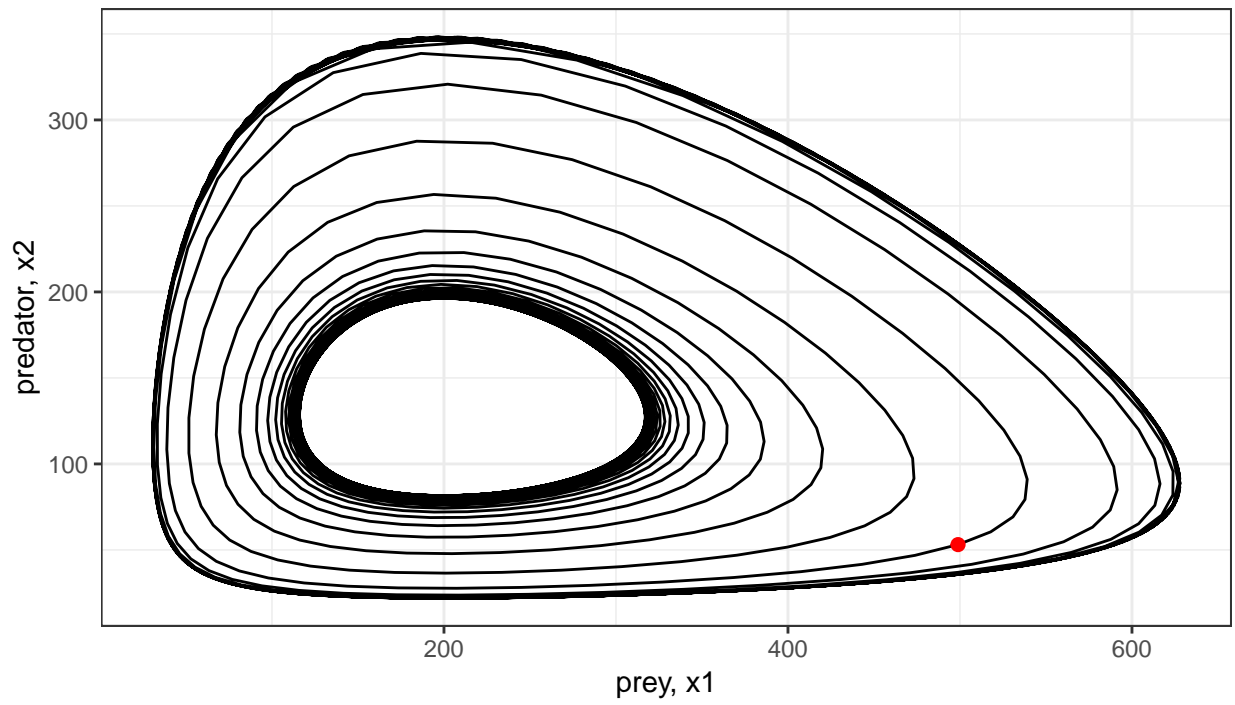


Figure 12: Phase space plot of system trajectory with regime shift. Approximate location of regime shift is indicated by red dot.

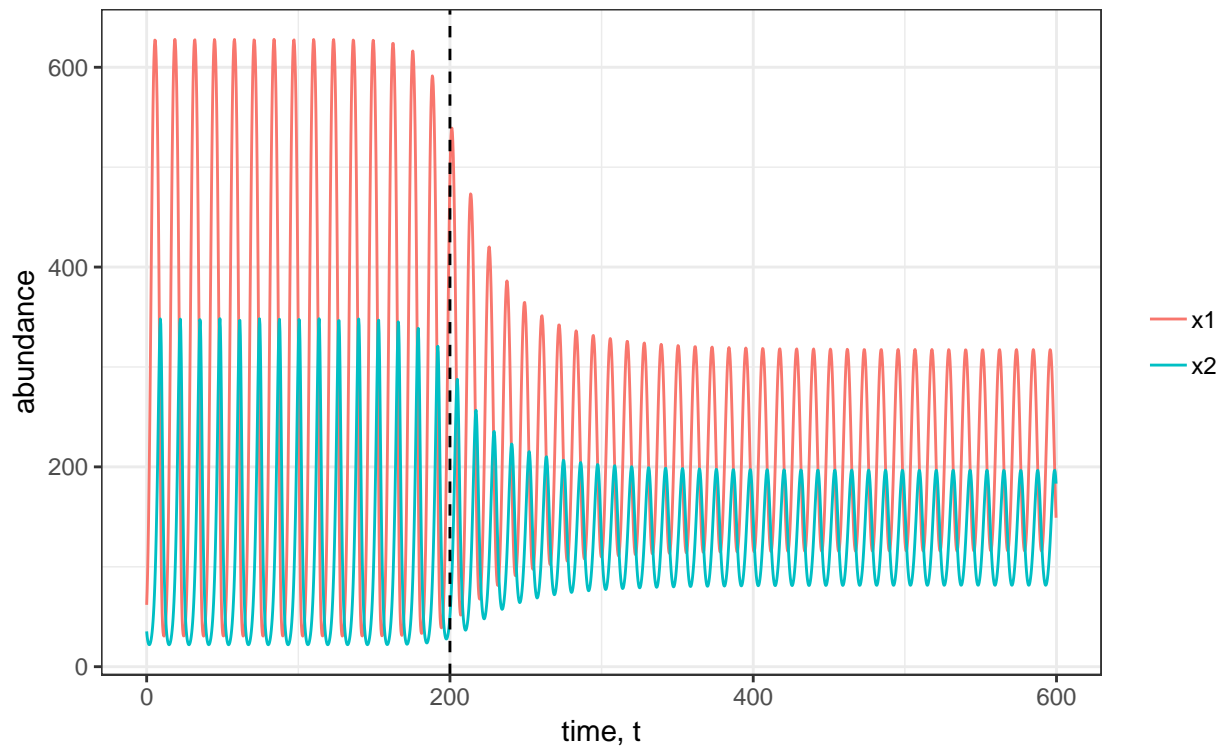


Figure 13: Predator prey abundance with regime shift

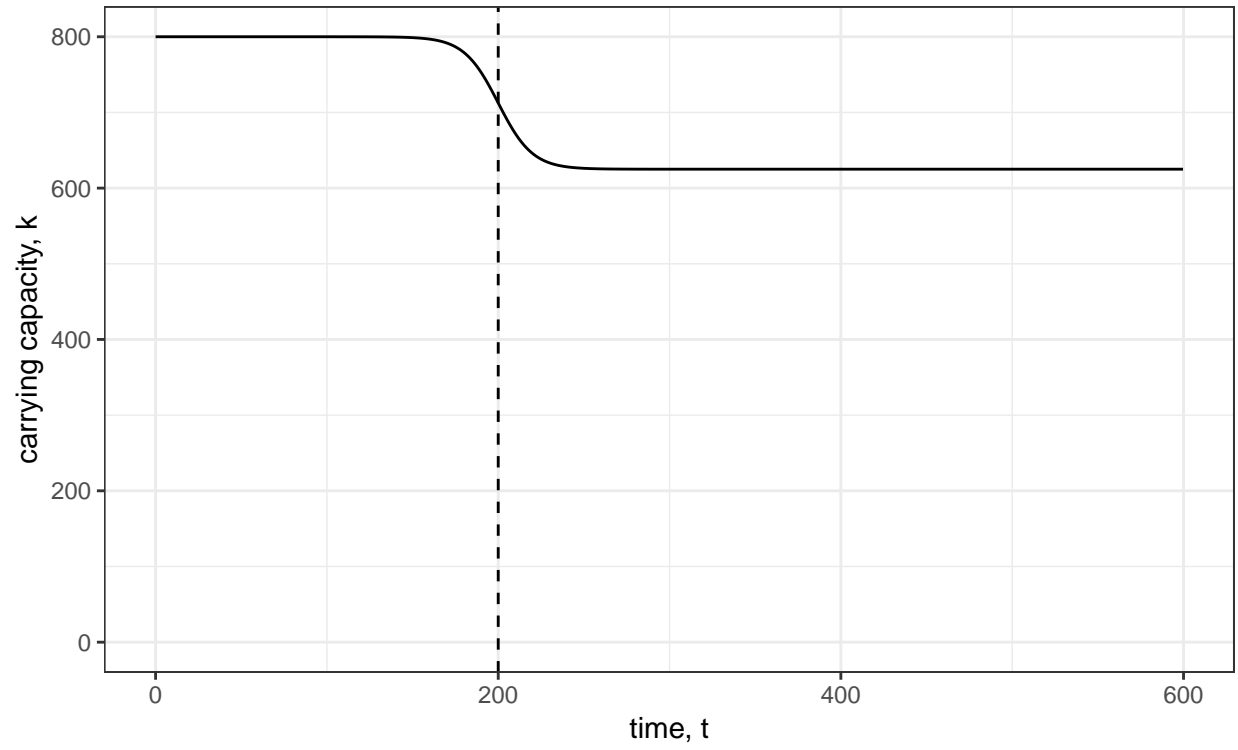


Figure 14: Carrying capacity with regime shift

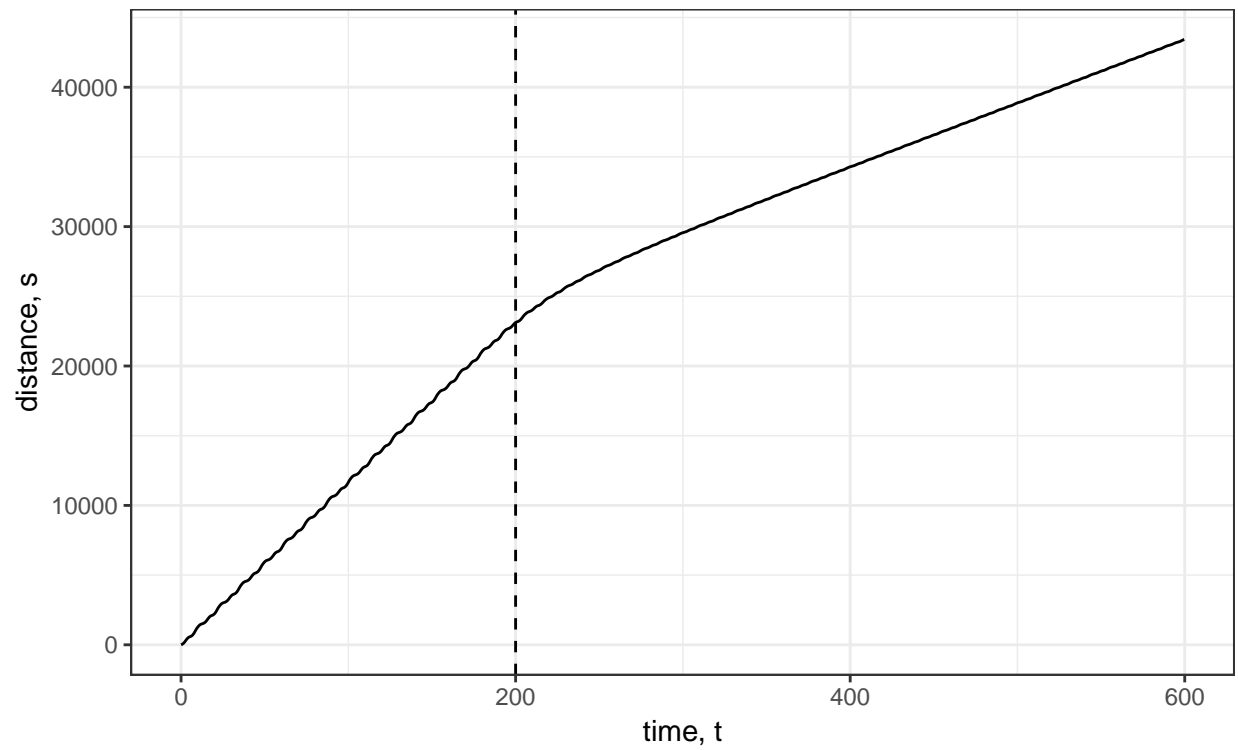


Figure 15: Distance traveled in phase space with regime shift

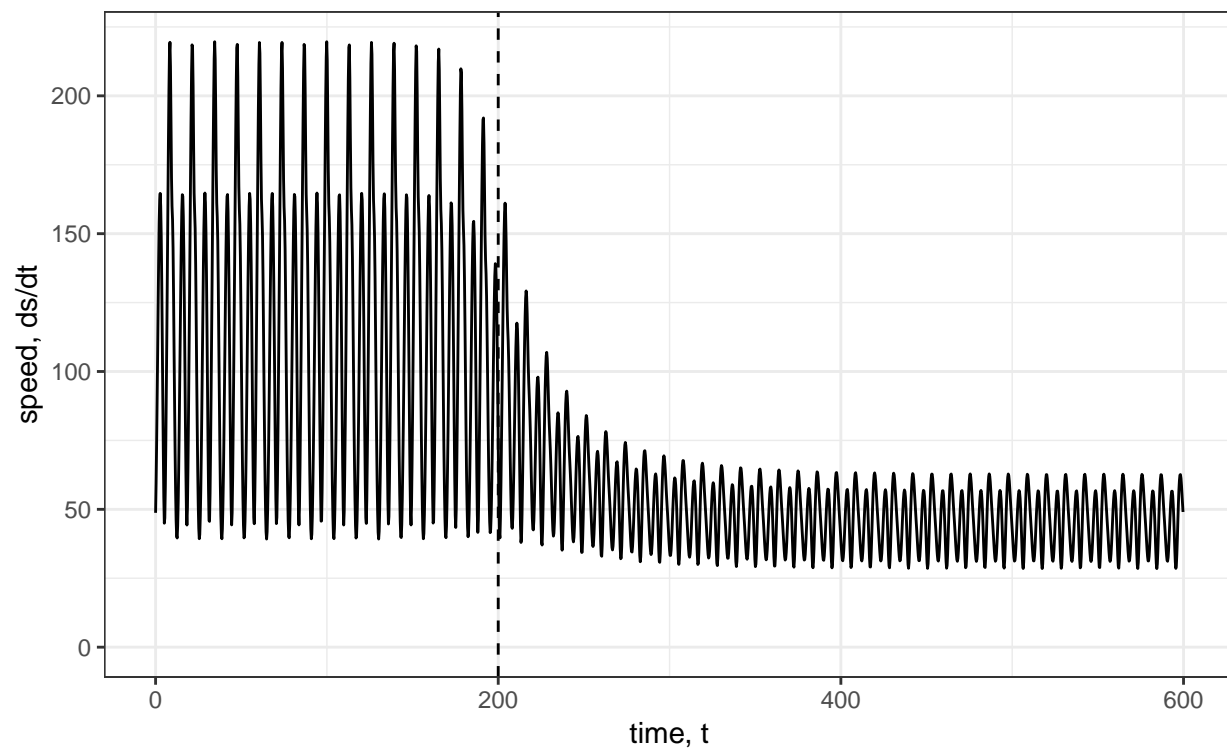


Figure 16: Speed with regime shift

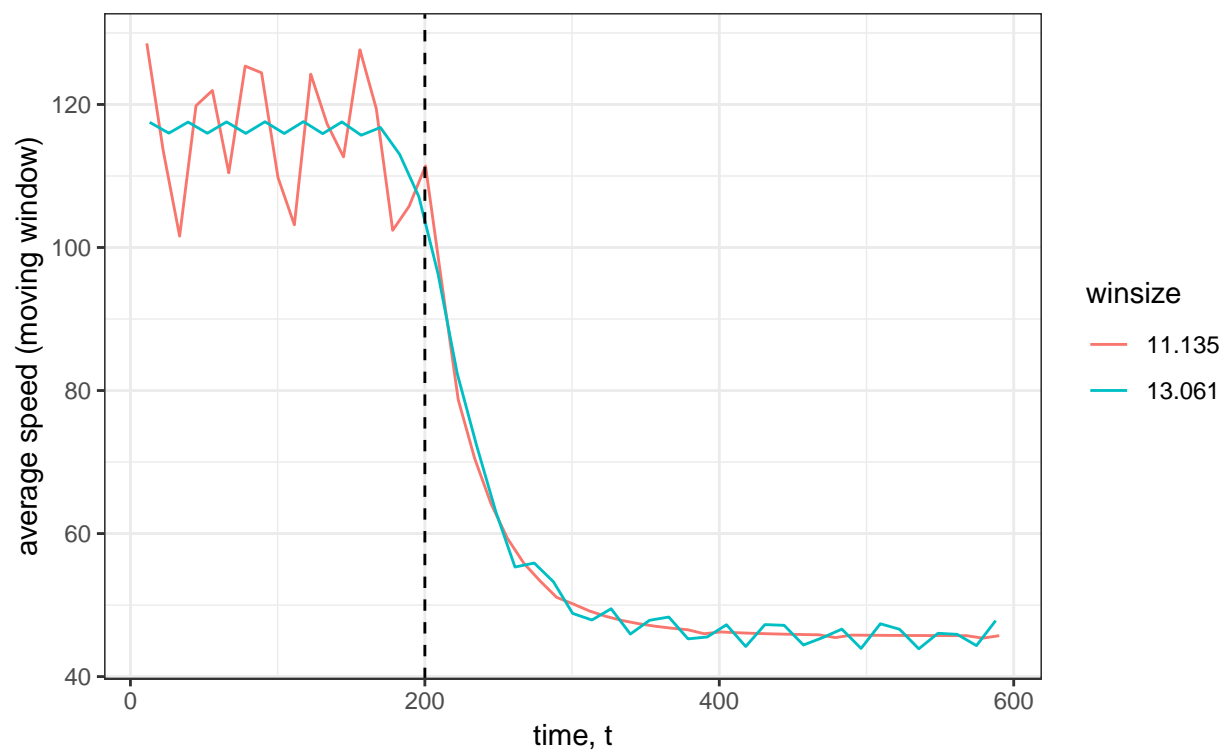


Figure 17: Average speed over moving window with regime shift

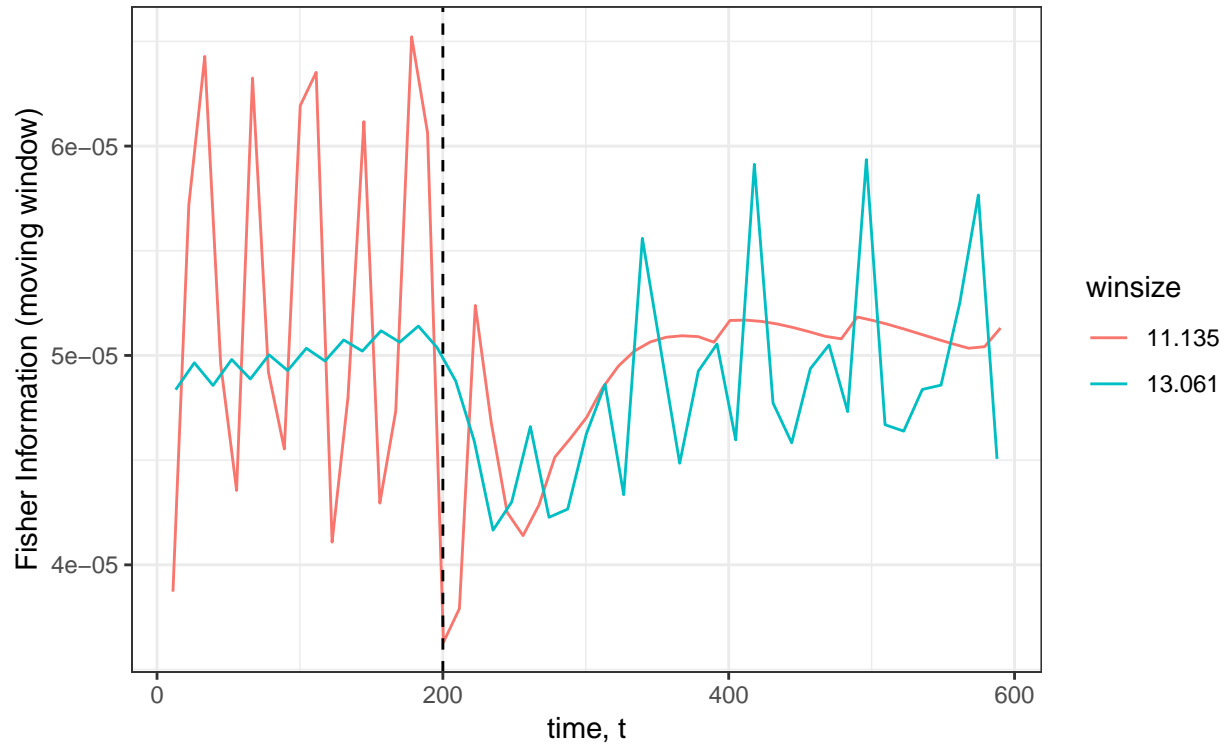


Figure 18: Fisher Information over moving window with regime shift

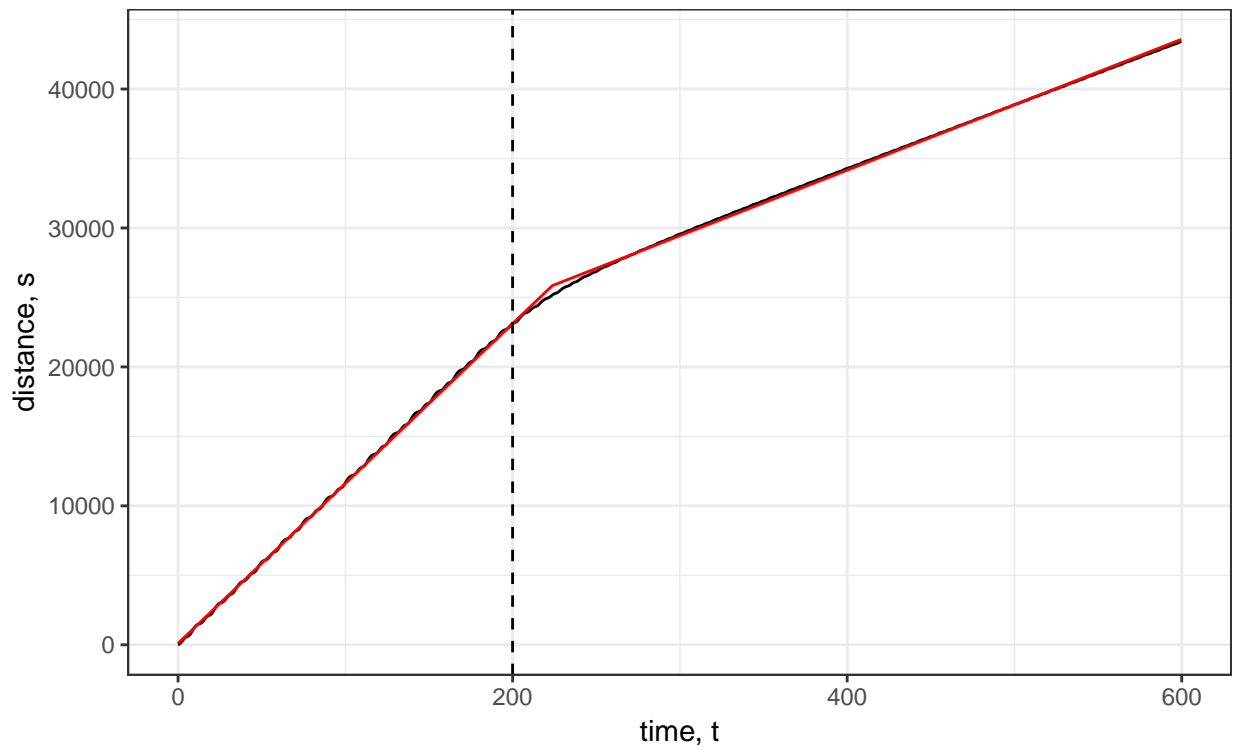


Figure 19: Piecewise linear fit to distance traveled

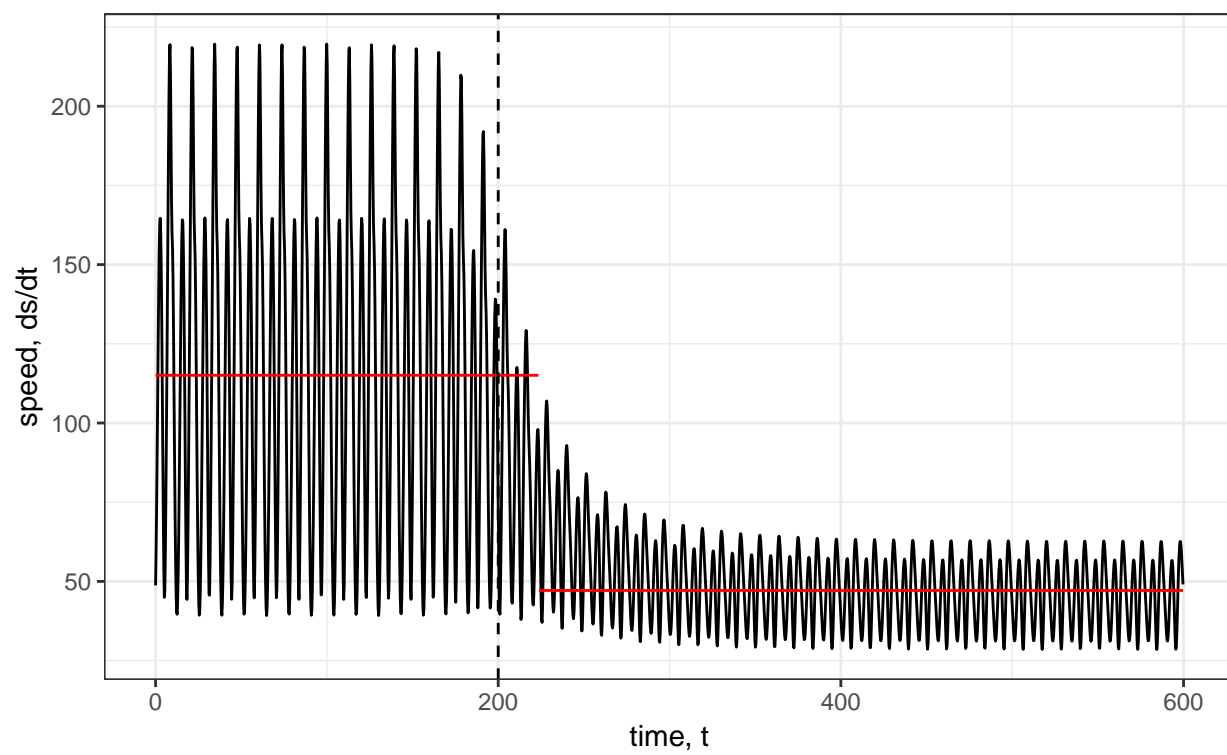


Figure 20: Speed based on piecewise linear fit to distance traveled