PE Model

Jordan Twombly Ellis

2025-06-24

PE Model instructions

Necessary packages to run model

```
library(dde)
library(deSolve)
library(tidyr)
library(ggplot2)
library(patchwork)
library(readr)
```

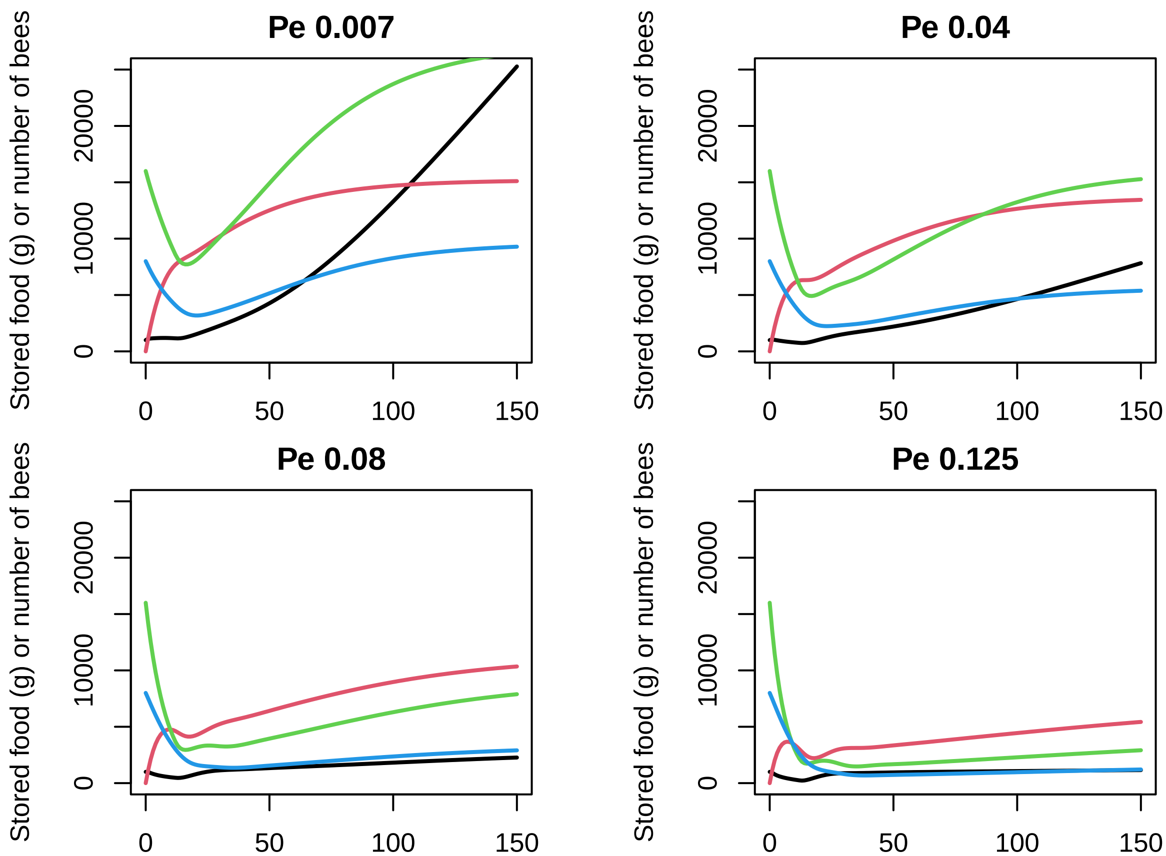
Initial model with MR set to 1

Figure 2 plots

```
bee_dde <- function(t, y, parms){
  L = 2000
  alphamin = 0.25
  sigma = 0.75
  v = 5000
  alphamax = 0.25
  phi = 1/9
  gammaA = 0.007
  gammaB = 0.018
  ct = 0.033
  b = 500
  a0 = 5
  pe = parms[1]
  mr= parms[2]
  delay = 12
  y1 = y[[1L]]
  y2 = y[[2L]]
  y3 = y[[3L]]
  y4 = y[[4L]]

  #tau = t - delay
  if(t < 12){
    y_lag = 0
  }
```

```r
  else{
    y_lag = deSolve::lagvalue(t - 12)[2]
  }

  R = alphamin + (alphamax*(b^2/(b^2+y1^2))) - (sigma*(y4)/(y4+y3))
  a = 1/R

  if(a <= 10){
    Na = -0.02*(a-10)^2+3.5
  }
  else{
    Na = -0.015*(a-10)^2 +3.5
  }

  if(a <= 40/3){
    Ta = 0.06 * (a-5) +0.5
  }
  else{
    Ta = 1
  }

  Ma = (((a-a0)^4) + 3) / ((4.94 + 0.08*a) * (a-a0)^4)

  dfdt <- (ct*Na*y4) - (gammaA * (y4 +y3)) - (gammaB*y2) # + cf
  dBdt <- (((L*y1^2) * y3) / ((b^2 + y1^2) * (y3+v))) - (phi*y2)
  dHdt <- (phi*y_lag) - (R * y3) - pe*y3
  dFdt <- (Ta * R * y3) - (mr * Ma * y4)

  list(c(dfdt,dBdt,dHdt,dFdt))
}

#Label timesteps for simulation to run over
tt = seq(0,150)

#Label initial values for model
y0 = c(y1 = 1000, y2 = 0, y3 = 16000, y4 = 8000)

#Create a premature hive exiting parameter, values used in manuscript: 0.007, 0.04, 0.08, and 0.125

peList = c(0.007, 0.04, 0.08, 0.125)

par(mfrow = c(2,2), mar=c(2,5,2,2))

for(i in 1:length(peList)){
  parms = c(pe = peList[i], mr = 1)

  #Run model and time it
yout = dede(y = y0, times = tt, func = bee_dde,
                         parms = parms)

#Plot output, pe values used in manuscript: 0.007, 0.04, 0.08, and 0.125
PE_plot = matplot(yout[,1], yout[,-1], type = "l", lwd = 2, lty = 1,
        main = paste("Pe", as.character(peList[i])), xlab = "Days",
```

```
        ylab = "Stored food (g) or number of bees",
        ylim = c(0,25000))

print(PE_plot)
}
```

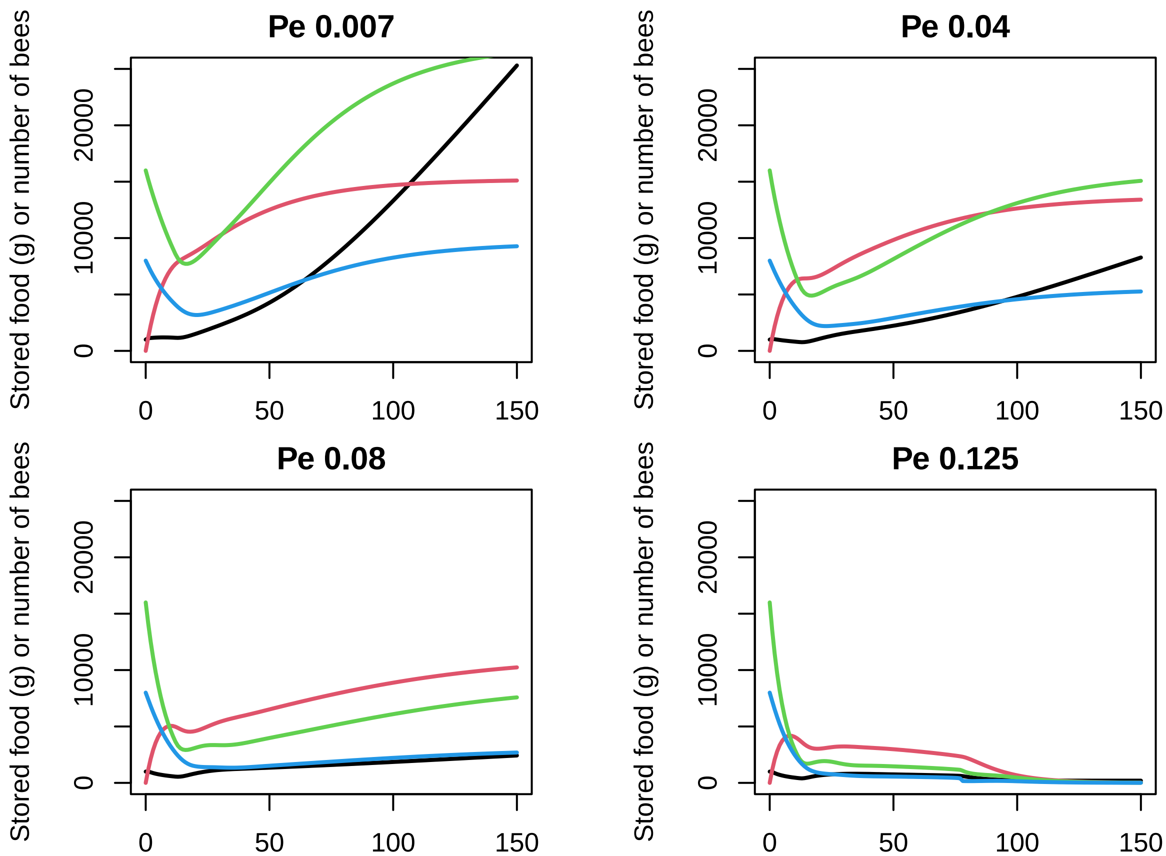


Model including equation linking PE and MR

Figure 3 plots

```
par(mfrow = c(2,2), mar=c(2,5,2,2))

for(i in 1:length(peList)){
  parms = c(pe = peList[i], mr = ((15*peList[i]^2) + 1))

  #Run model and time it
yout = dede(y = y0, times = tt, func = bee_dde,
                         parms = parms)

#Plot output, pe values used in manuscript: 0.007, 0.04, 0.08, and 0.125
PE_plot = matplot(yout[,1], yout[,-1], type = "l", lwd = 2, lty = 1,
        main = paste("Pe", as.character(peList[i])), xlab = "Days",
        ylab = "Stored food (g) or number of bees",
        ylim = c(0,25000))
```

```r
print(PE_plot)
}
```

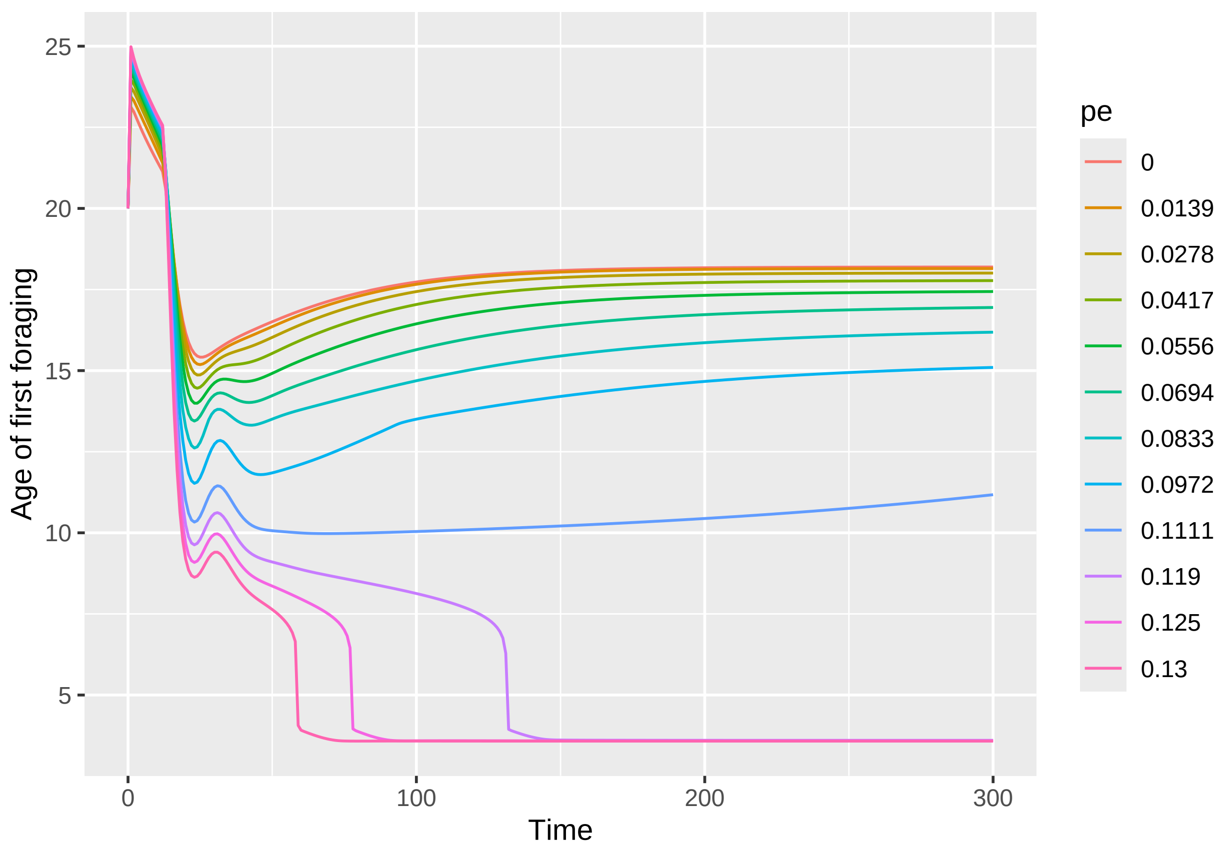


Age of first foraging at different PE levels

Figure 4

```r
#Create a function to repeatedly run simulations with different pe params using lapply
run_many_sims <- function(pe){
  mr = ((15*pe^2) + 1)
  yout <- dede(y = c(y1 = 1000, y2 = 0, y3 = 16000, y4 = 8000),
               times = seq(0,300),
               func = bee_dde,
               parms = c(pe, mr))
  colnames(yout) = c("time","food", "brood", "hive", "foragers")
  yout = data.frame(yout)
  yout$pe = pe
  yout
}

#Choose range of self removal rates to simulate
removal_rates <- seq(from = 0, to = 0.125, length.out = 10)
removal_rates <- c(removal_rates, 0.119, 0.13)
```

```r
#Create function to get foraging age of list of simulations
get_foraging_age <- function(df){
  ageForaging <- 0.25 +(0.25*(500^2/(500^2+df$food^2))) - (0.75*(df$foragers/(df$foragers + df$hive)))
  ageForaging <- 1/ageForaging
  ages <- data.frame(time = df$time, age = ageForaging, pe = df$pe)
}

simResults <- lapply(X = removal_rates, FUN = run_many_sims)
ages <- lapply(X = simResults, FUN = get_foraging_age)

allAges <- data.frame(mapply(c, ages[[1]], ages[[2]], ages[[3]], ages[[4]], ages[[5]],ages[[6]], ages[[

allAges$pe <- round(allAges$pe, 4)
allAges$pe = factor(allAges$pe)

ggplot(allAges, aes(x = time, y = age, col = pe))+
  geom_line()+
  ylab("Age of first foraging")+
  xlab("Time")
```



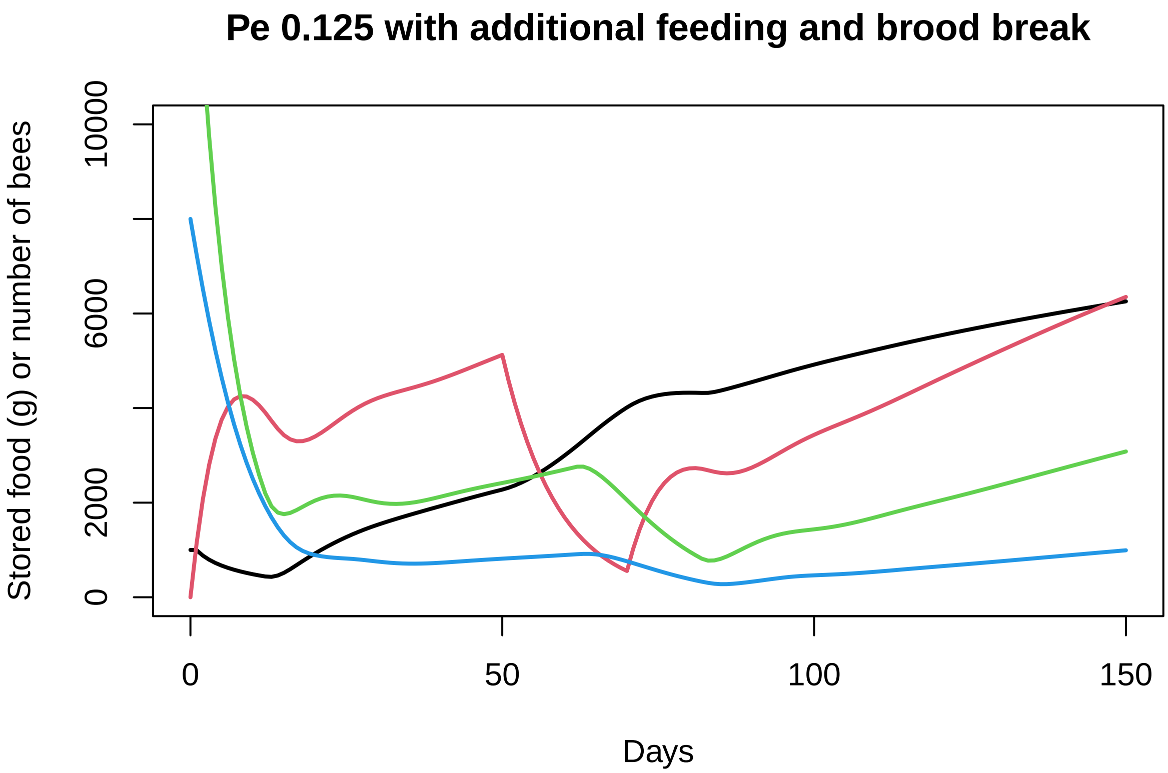
Model with added feeding and a brood break

This is an example implementing both brood break and feeding as shown in figure 5 panel d. This should give an idea of how to create all Fig. 5 plots

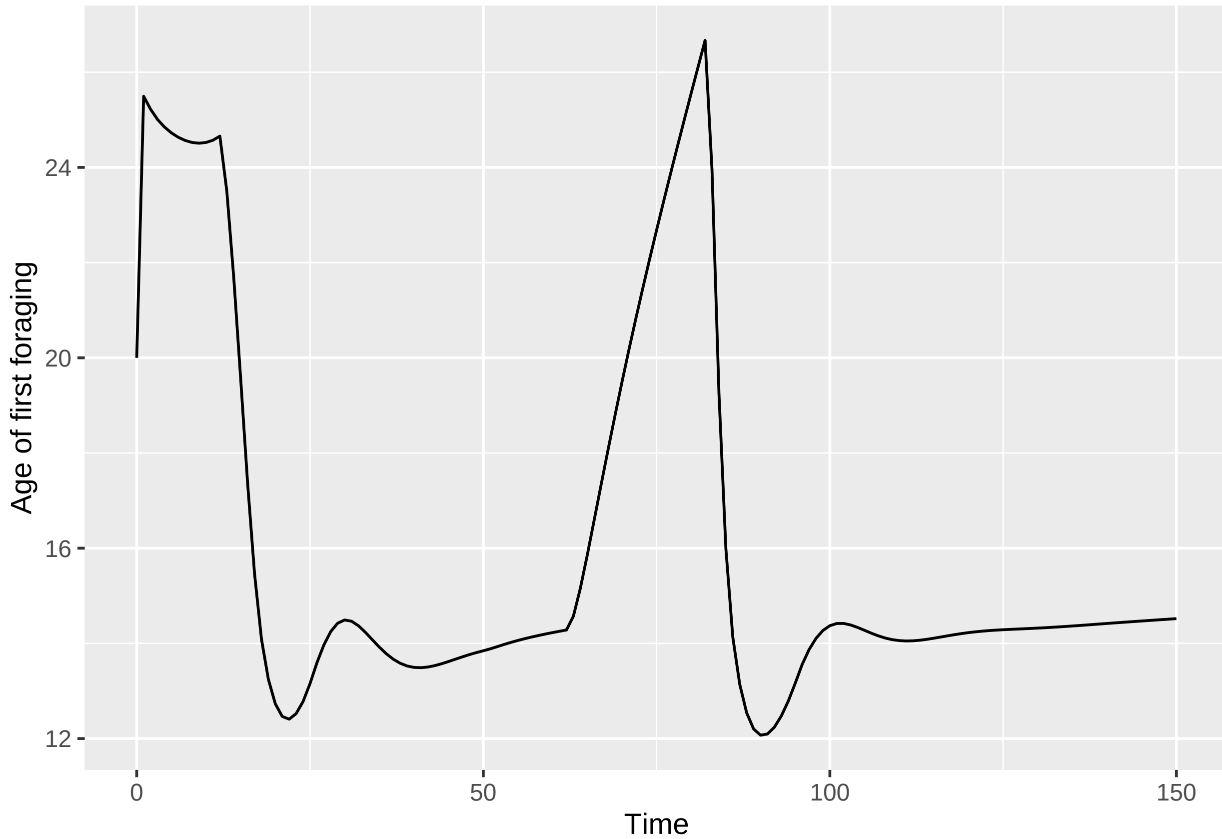
Bottom plot is example of plotting age of first foraging with brood break and feeding, as shown in figure 6 panel d

```
bee_dde <- function(t, y, parms){
  #Implementation of brood break
  if (t < 50) {
    L = 2000
  } else if (t < 70) {
    L = 0
  } else{
    L = 2000
  }
  alphamin = 0.25
  sigma = 0.75
  v = 5000
  alphamax = 0.25
  phi = 1 / 9
  gammaA = 0.007
  gammaB = 0.018
  ct = 0.033
  b = 500
  a0 = 5
  #Addition of feeding parameter cf
  cf = 60
  pe = parms[1]
  mr= (15*pe^2) + 1
  delay = 12
  y1 = y[[1L]]
  y2 = y[[2L]]
  y3 = y[[3L]]
  y4 = y[[4L]]

  #tau = t - delay
  if(t < 12){
    y_lag = 0
  }

  else{
    y_lag = deSolve::lagvalue(t - 12)[2]
  }

  R = alphamin + (alphamax*(b^2/(b^2+y1^2))) - (sigma*(y4)/(y4+y3))
  a = 1/R

  if(a <= 10){
    Na = -0.02*(a-10)^2+3.5
  }
  else{
    Na = -0.015*(a-10)^2 +3.5
  }

  if(a <= 40/3){
    Ta = 0.06 * (a-5) +0.5
  }
```

```
  else{
    Ta = 1
  }

  Ma = (((a-a0)^4) + 3) / ((4.94 + 0.08*a) * (a-a0)^4)

  dfdt <- (ct*Na*y4) - (gammaA * (y4 +y3)) - (gammaB*y2) + cf
  dBdt <- (((L*y1^2) * y3) / ((b^2 + y1^2) * (y3+v))) - (phi*y2)
  dHdt <- (phi*y_lag) - (R * y3) - pe*y3
  dFdt <- (Ta * R * y3) - (mr * Ma * y4)

  list(c(dfdt,dBdt,dHdt,dFdt))
}

#Label timesteps for simulation to run over
tt = seq(0,150)

#Label initial values for model
y0 = c(y1 = 1000, y2 = 0, y3 = 16000, y4 = 8000)

#Premature hive exiting parameter set to 0.125
parms = c(pe = 0.125)

#Run model and time it
yout = dede(y = y0, times = tt, func = bee_dde,
                         parms = parms)

#Plot output, pe values used in manuscript: 0.007, 0.04, 0.08, and 0.125
matplot(yout[,1], yout[,-1], type = "l", lwd = 2, lty = 1,
        main = "Pe 0.125 with additional feeding and brood break", xlab = "Days",
        ylab = "Stored food (g) or number of bees",
        ylim = c(0,10000))
```




```r
colnames(yout) = c("time","food", "brood", "hive", "foragers")
yout = data.frame(yout)
yout$pe = 0.125
ages = get_foraging_age(yout)


ggplot(ages, aes(x = time, y = age))+
  geom_line()+
  ylab("Age of first foraging")+
  xlab("Time")
```



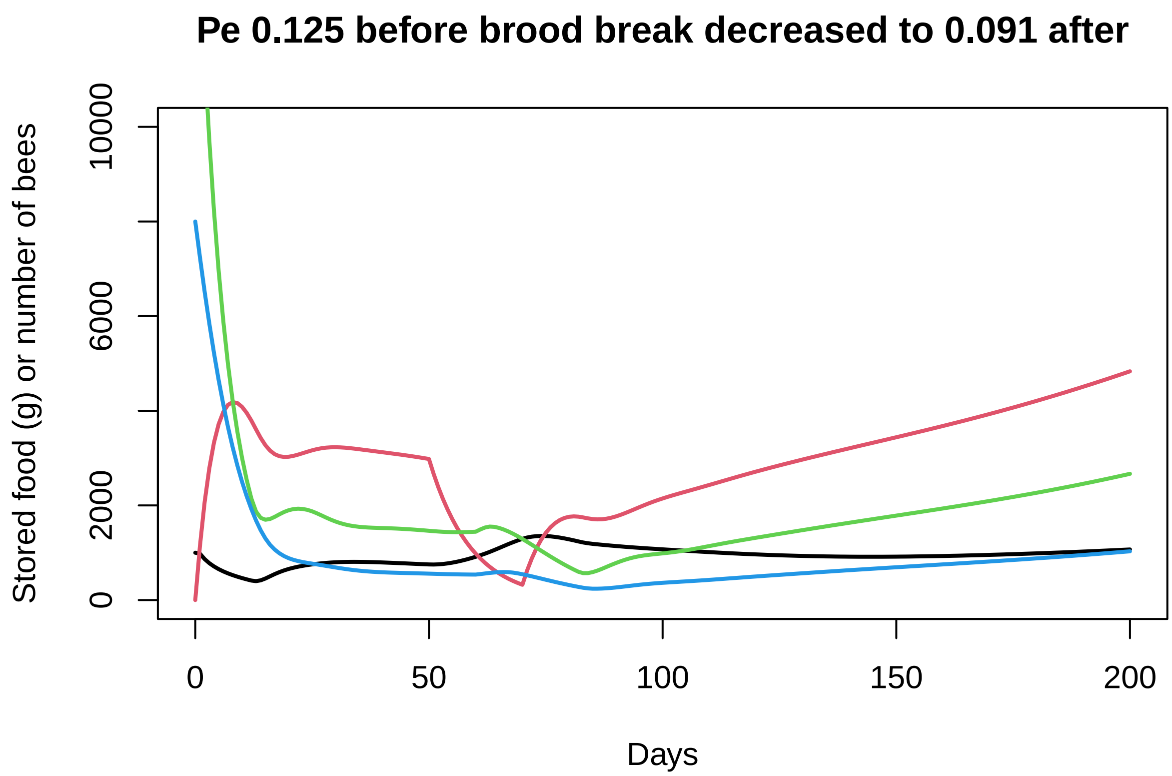
Model with PE decrease after the brood break

Figure 7 plot panel c

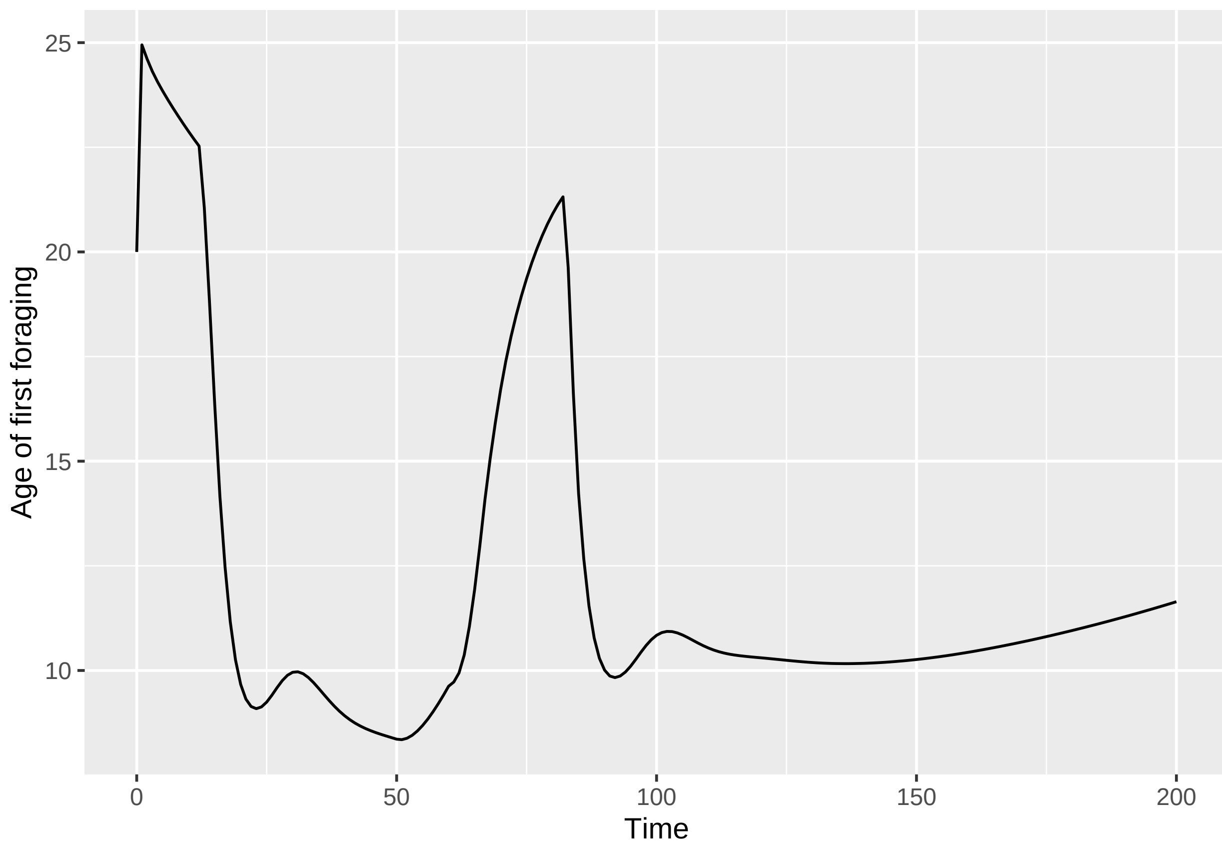
```
bee_dde <- function(t, y, parms){
  #Implementation of brood break
  if (t < 50) {
    L = 2000
  } else if (t < 70) {
    L = 0
  } else{
    L = 2000
  }
  alphamin = 0.25
  sigma = 0.75
  v = 5000
  alphamax = 0.25
  phi = 1/9
  gammaA = 0.007
  gammaB = 0.018
  ct = 0.033
  b = 500
  a0 = 5
  #Setting pe to decrease 25% after brood break
  if (t < 60) {
```

```
    pe = parms[1]
  } else{
    pe = parms[2]
  }
  mr= (15*pe^2) + 1
  delay = 12
  y1 = y[[1L]]
  y2 = y[[2L]]
  y3 = y[[3L]]
  y4 = y[[4L]]

  #tau = t - delay
  if(t < 12){
    y_lag = 0
  }

  else{
    y_lag = deSolve::lagvalue(t - 12)[2]
  }

  R = alphamin + (alphamax*(b^2/(b^2+y1^2))) - (sigma*(y4)/(y4+y3))
  a = 1/R

  if(a <= 10){
    Na = -0.02*(a-10)^2+3.5
  }
  else{
    Na = -0.015*(a-10)^2 +3.5
  }

  if(a <= 40/3){
    Ta = 0.06 * (a-5) +0.5
  }
  else{
    Ta = 1
  }

  Ma = (((a-a0)^4) + 3) / ((4.94 + 0.08*a) * (a-a0)^4)

  dfdt <- (ct*Na*y4) - (gammaA * (y4 +y3)) - (gammaB*y2)
  dBdt <- (((L*y1^2) * y3) / ((b^2 + y1^2) * (y3+v))) - (phi*y2)
  dHdt <- (phi*y_lag) - (R * y3) - pe*y3
  dFdt <- (Ta * R * y3) - (mr * Ma * y4)

  list(c(dfdt,dBdt,dHdt,dFdt))
}

#Label timesteps for simulation to run over
tt = seq(0,200)

#Label initial values for model
y0 = c(y1 = 1000, y2 = 0, y3 = 16000, y4 = 8000)
```

```r
#Premature hive exiting parameter before and after brood break
parms = c(pe_before = 0.125, pe_after = 0.091)

#Run model and time it
yout = dede(y = y0, times = tt, func = bee_dde,
                         parms = parms)

#Plot output, pe values used in manuscript: 0.007, 0.04, 0.08, and 0.125
matplot(yout[,1], yout[,-1], type = "l", lwd = 2, lty = 1,
        main = "Pe 0.125 before brood break decreased to 0.091 after", xlab = "Days",
        ylab = "Stored food (g) or number of bees",
        ylim = c(0,10000))
```




```r
colnames(yout) = c("time","food", "brood", "hive", "foragers")
yout = data.frame(yout)
yout$pe = 0.125
ages = get_foraging_age(yout)


ggplot(ages, aes(x = time, y = age))+
  geom_line()+
  ylab("Age of first foraging")+
  xlab("Time")
```


Age of first foraging
25
20
15
10
0
50
100
150
200
Time