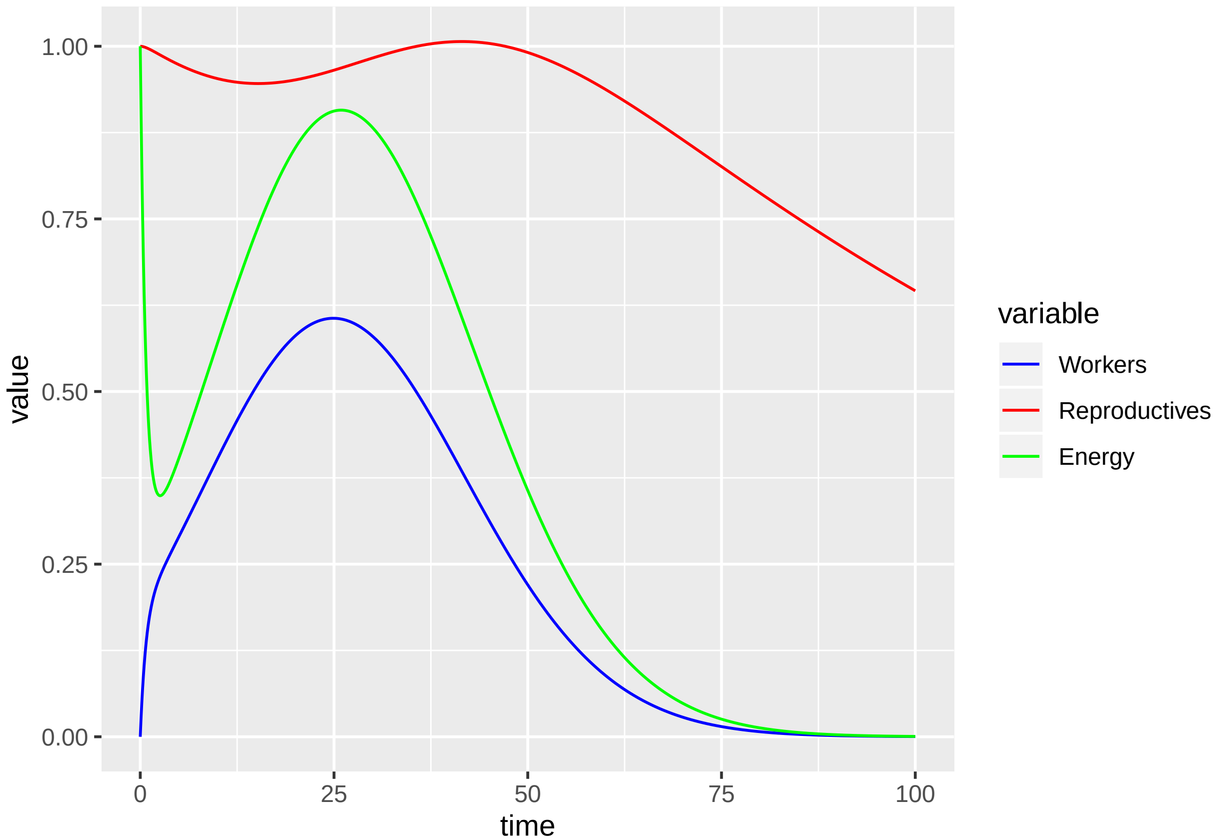
Numerical Solution of Systems of Differential Equations

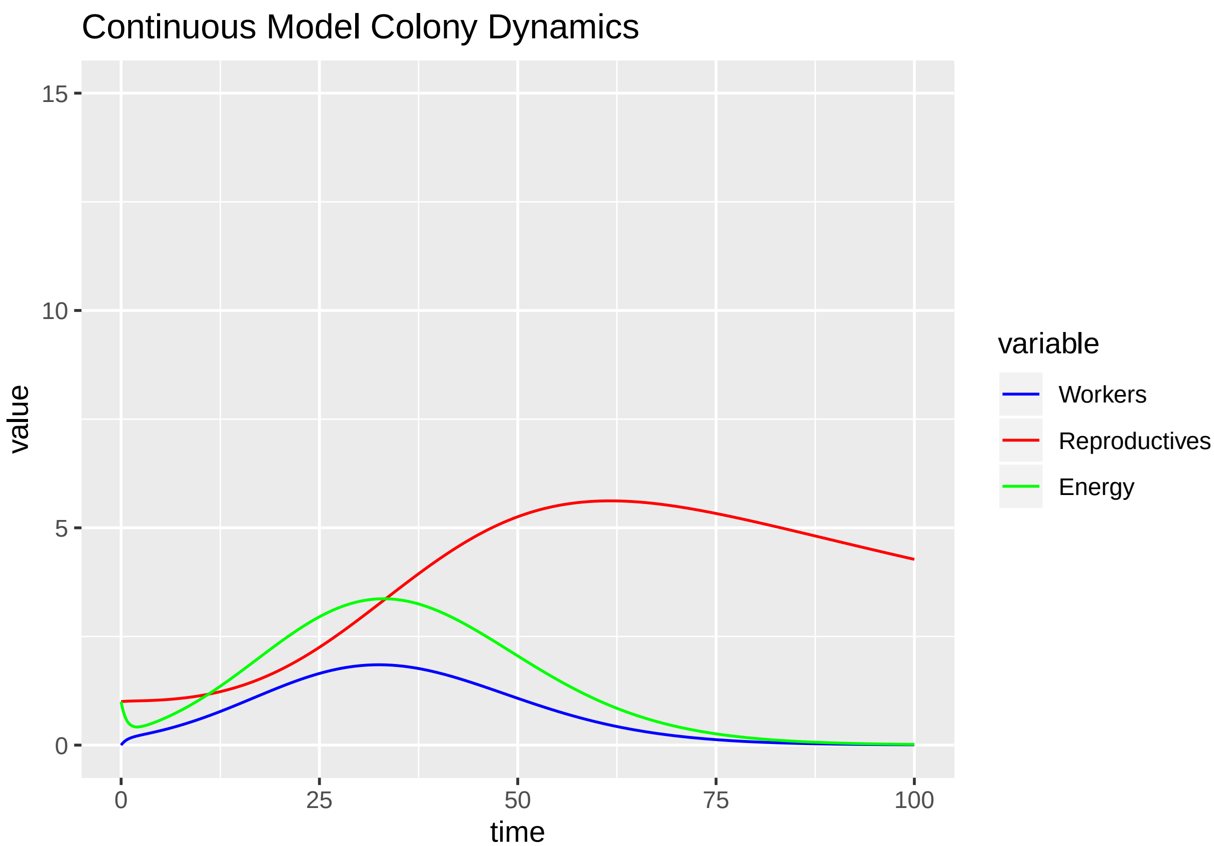
```r
library(mosaic)
library(mosaicCalc)
library(deSolve)
library(ggplot2)
library(tidyr)
library(RColorBrewer)
library(deSolve)
library(reshape2)
```

```r
env <- c( 0.4 , 0.4 , 0.95 , 0.7)
col <- c( 0.1 , 0.1 , .2 , 0.3)
state <- c(Workers = 0,Reproductives = 1, Energy = 1)
parameters <- c(Dw1 = env[1], Dw2 = env[2], Dw3 = env[3], Dw4 = env[4],
                Br1 = col[1], Br2 = col[2], Br3 = col[3], Br4 = col[4],
                Cw = 3.5, Cr = 3, Kw = 1.5, Kr = 0, Pe = 1, Dr = .01, Ttol = 100)

times <- seq(0, 100, by = .1)

systemDE <- function(t, state, parameters){
  with(as.list(c(state, parameters)),{
          tt <- t/Ttol
          Dw <- Dw4*(Dw1*(tt-1)^2 + Dw2*(1-4*(tt-.5)^2) + Dw3*tt^2)
          Br <- Br4*(Br1*(tt-1)^2 + Br2*(1-4*(tt-.5)^2) + Br3*tt^2)

          dWorkers <- -Dw*Workers+(1-Br)*Energy*Pe/Cw
          dReproductives <- Br*Energy*Pe/Cr-Dr*Reproductives
          dEnergy <- Kw*Workers+Kr*Reproductives-Pe*Energy
          list(c(dWorkers, dReproductives, dEnergy))
})}

DEsol <- ode(y = state, times = times, func = systemDE, parms = parameters)

dfDEsol <- melt(as.data.frame(DEsol), id.vars = "time")

ggplot(dfDEsol, mapping = aes(x=time,y=value, color=variable)) + geom_line() +
  scale_colour_manual(values=c("blue", "red", "green"))
```




```
env <- c( 0.4 , 0.4 , 0.9, 0.7)
col <- c( 0.1 , 0.1 , .2 , 1)
Kf0 <- c(   1 ,  .4 ,  1 , 2)
state <- c(Workers = 0,Reproductives = 1, Energy = 1)
parameters <- c(Dw1 = env[1], Dw2 = env[2], Dw3 = env[3], Dw4 = env[4],
                Br1 = col[1], Br2 = col[2], Br3 = col[3], Br4 = col[4],
                Kf1 = Kf0[1], Kf2 = Kf0[2], Kf3 = Kf0[3], Kf4 = Kf0[4],
                Cw = 3.5, Cr = 3, Pe = 1, Dr = .01, Kr = 0, Ttol = 100)

times <- seq(0, 100, by = .1)

systemDE <- function(t, state, parameters){
  with(as.list(c(state, parameters)),{
          tt <- t/Ttol
          Dw <- Dw4*(Dw1*(tt-1)^2 + Dw2*(1-4*(tt-.5)^2) + Dw3*tt^2)
          Br <- Br4*(Br1*(tt-1)^2 + Br2*(1-4*(tt-.5)^2) + Br3*tt^2)
          Kw <- Kf4*(Kf1*(tt-1)^2 + Kf2*(1-4*(tt-.5)^2) + Kf3*tt^2)

          dWorkers <- -Dw*Workers+(1-Br)*Energy*Pe/Cw
          dReproductives <- Br*Energy*Pe/Cr-Dr*Reproductives
          dEnergy <- Kw*Workers+Kr*Reproductives-Pe*Energy

          list(c(dWorkers, dReproductives, dEnergy))
})}

DEsol <- ode(y = state, times = times, func = systemDE, parms = parameters)
```

```r
dfDEsol <- melt(as.data.frame(DEsol), id.vars = "time")

ggplot(dfDEsol, mapping = aes(x=time,y=value, color=variable)) + geom_line() + ylim(0,15) +
  scale_colour_manual(values=c("blue", "red", "green")) + ggtitle("Continuous Model Colony Dynamics")
```




```r
n <- 10
param.vals <- seq(from=0,to=1,length.out = n)
dfDEsol2 <- list(); p <- list();
temp <- ode(y = state, times = times, func = systemDE, parms = parameters)
dfDEsol3 <- melt(as.data.frame(DEsol), id.vars = "time")[,c(1,2)]

for (i in 1:n)
{
  parameters["Dr"] <- param.vals[i]
  DEsol2 <- ode(y = state, times = times, func = systemDE, parms = parameters)
  dfDEsol2[[i]] <- melt(as.data.frame(DEsol2), id.vars = "time")
  dfDEsol3 <- cbind(dfDEsol3,dfDEsol2[[i]][,3])
}
names(dfDEsol3) <- c("t","var",as.character(1:n))
forplot <- melt(dfDEsol3,id.vars = c("t","var"))
forplot[1000:1005,];forplot[3000:3010,]
```

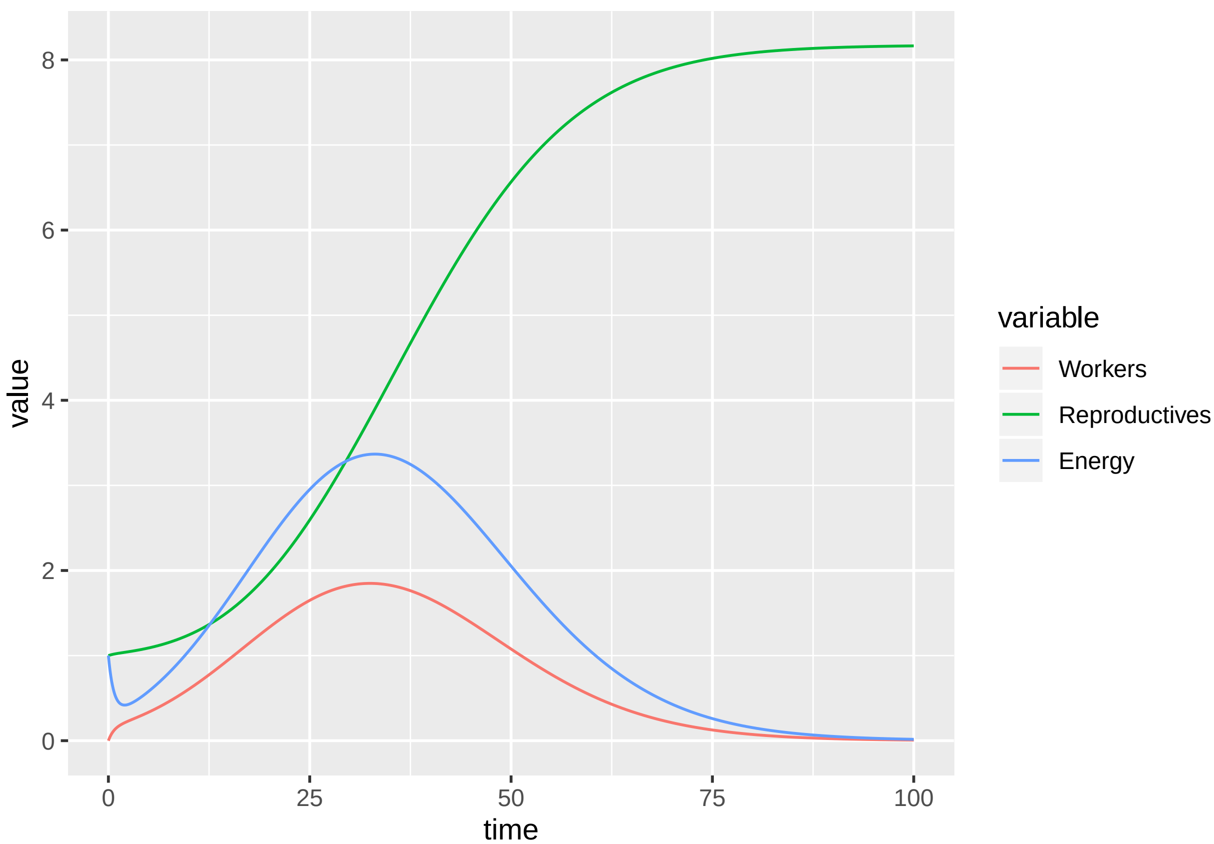
```
##          t           var variable       value
## 1000  99.9       Workers        1 0.007328121
## 1001 100.0       Workers        1 0.007244082
## 1002   0.0 Reproductives        1 1.000000000
## 1003   0.1 Reproductives        1 1.003177879
```

```
## 1004   0.2 Reproductives        1 1.006074419
## 1005   0.3 Reproductives        1 1.008729477
```

```
##          t     var variable      value
## 3000  99.7  Energy        1 0.01686826
## 3001  99.8  Energy        1 0.01668057
## 3002  99.9  Energy        1 0.01649503
## 3003 100.0  Energy        1 0.01631163
## 3004   0.0 Workers        2 0.00000000
## 3005   0.1 Workers        2 0.02414263
## 3006   0.2 Workers        2 0.04543024
## 3007   0.3 Workers        2 0.06425071
## 3008   0.4 Workers        2 0.08093962
## 3009   0.5 Workers        2 0.09578728
## 3010   0.6 Workers        2 0.10904561
```

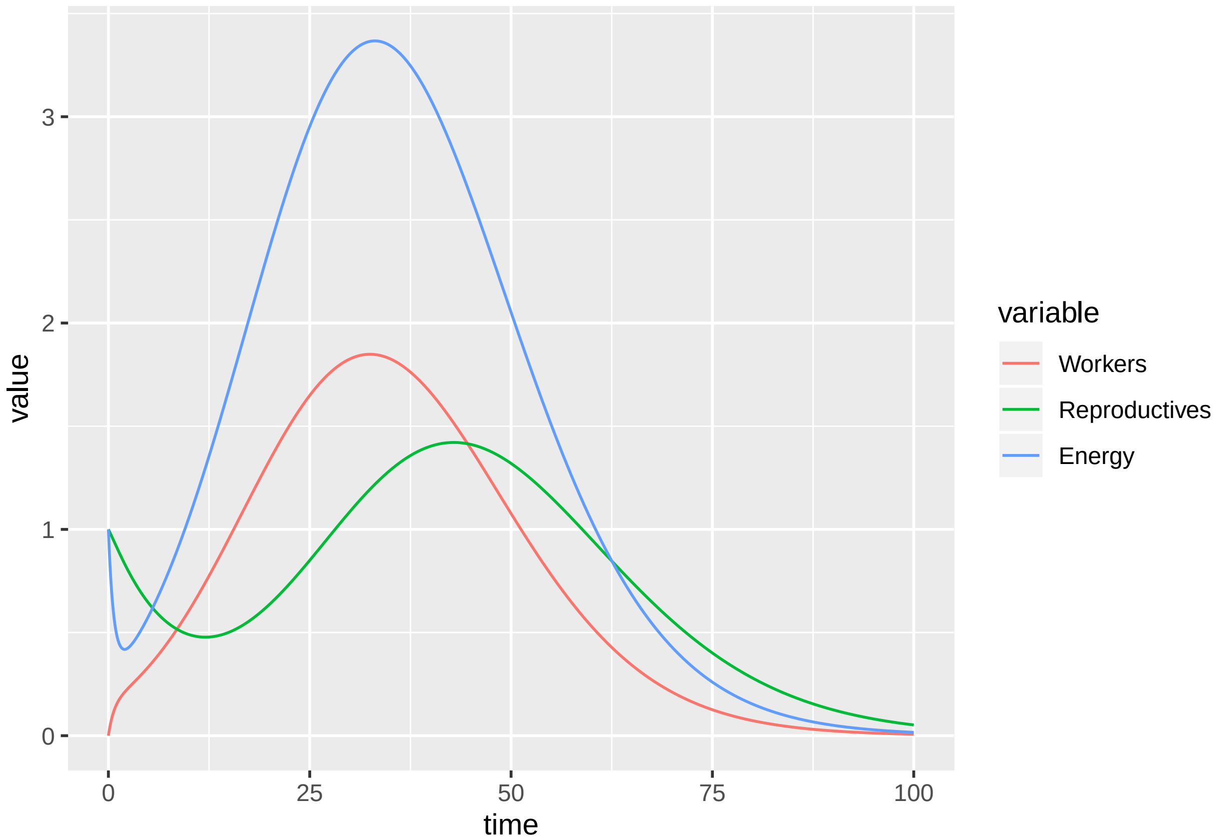
```
#ggplot(forplot) + geom_line(aes(x=t,y=value,group=var,color=var))
#dfDEsol3
#melt(dfDEsol3,id.vars = c("t","var"))

for(j in 1:n)
{
  p[[j]] <- ggplot(dfDEsol2[[j]]) + geom_line(mapping = aes(x=time,y=value,color=variable))
}
p
```

```
## [[1]]
```



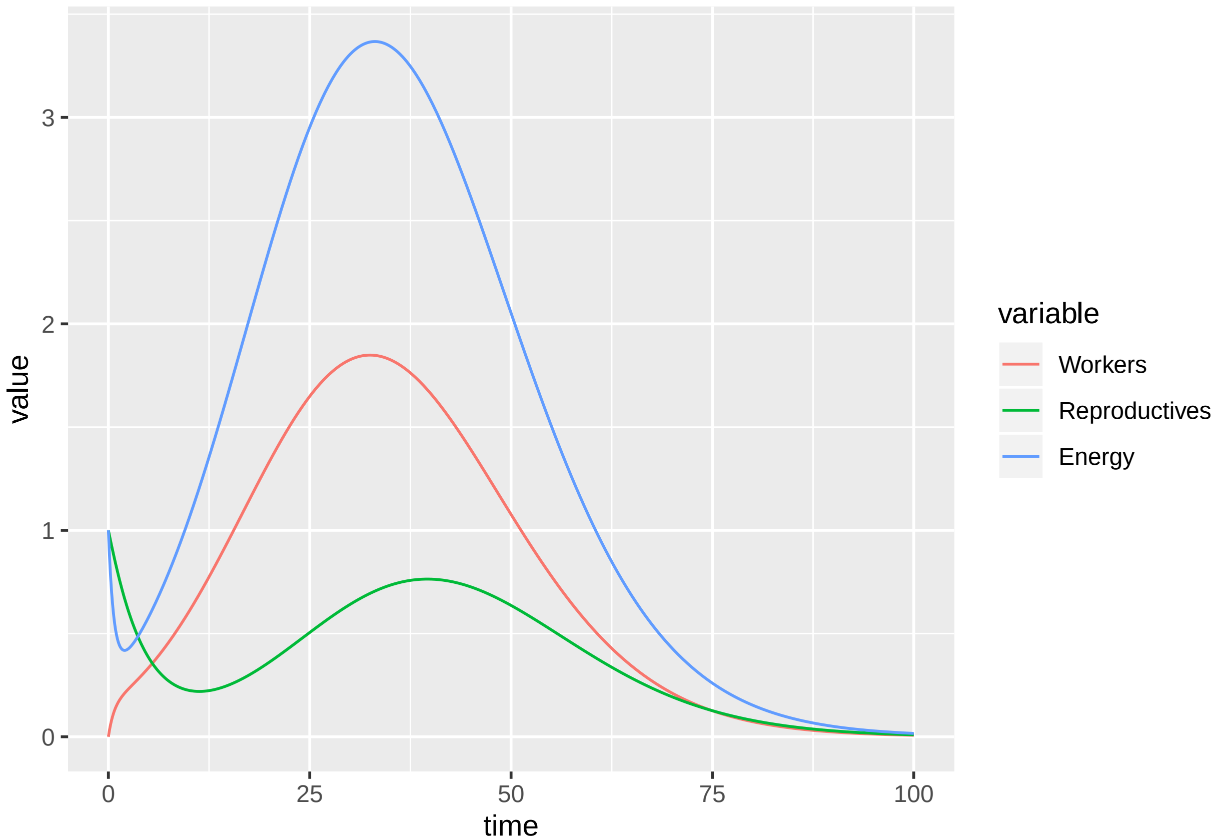

```
##
```

```
## [[2]]
```



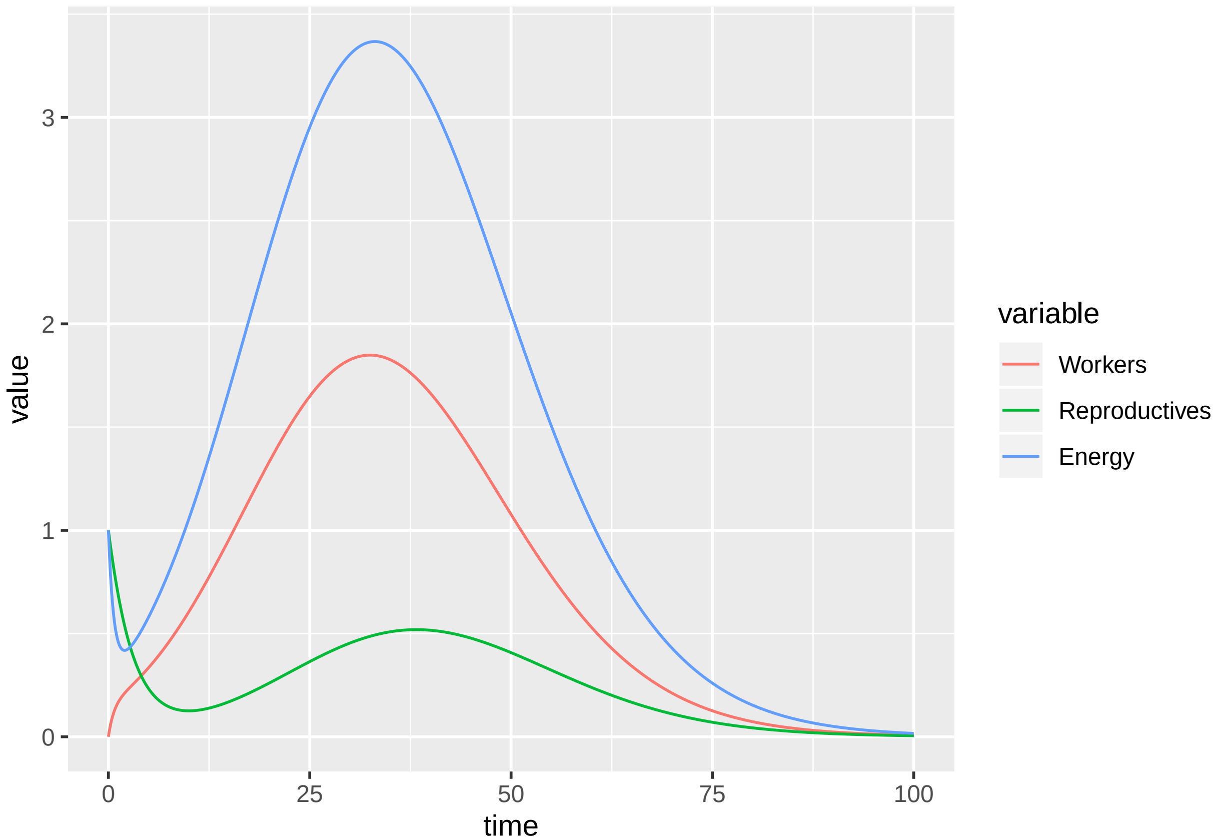
value
3
2
1
0
0
25
50
75
100
time
variable
Workers
Reproductives
Energy


```
##
## [[3]]
```



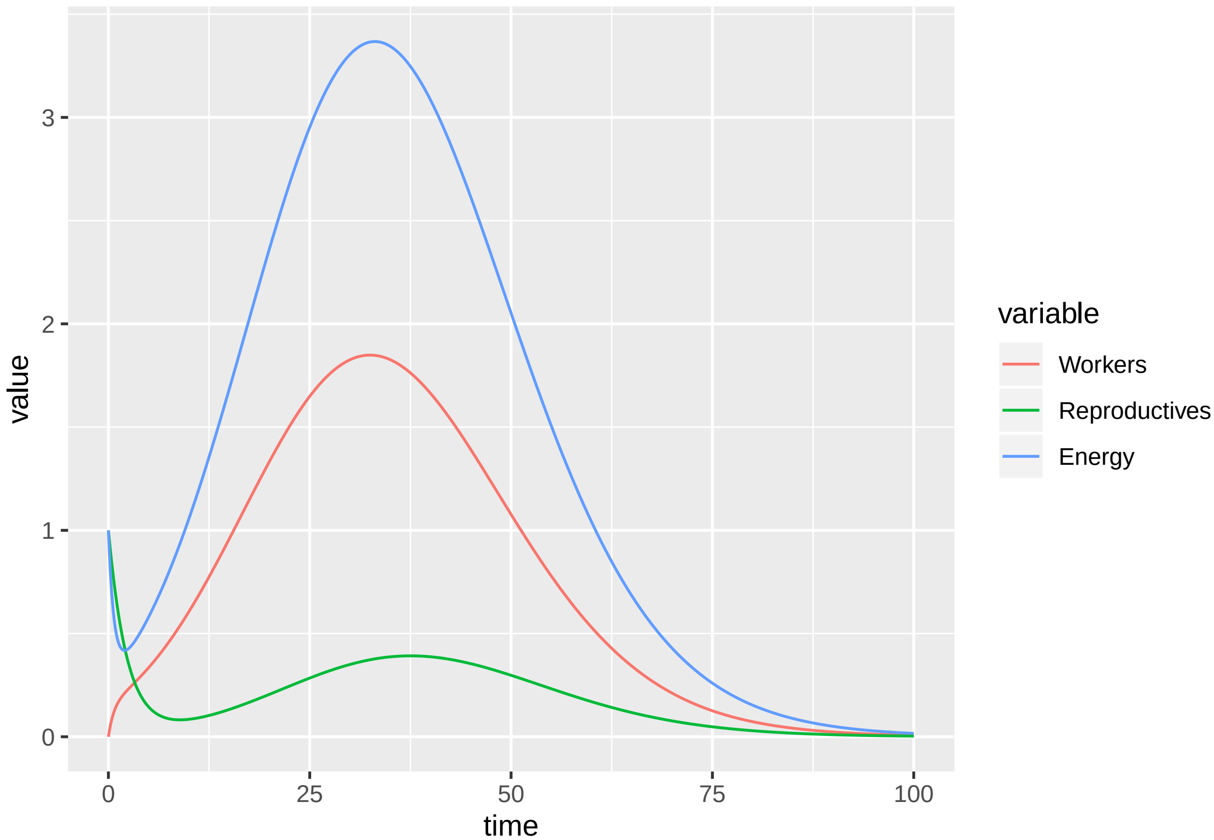
3
2
1
0
value
0
25
50
75
100
time
variable
Workers
Reproductives
Energy


```
##
## [[4]]
```



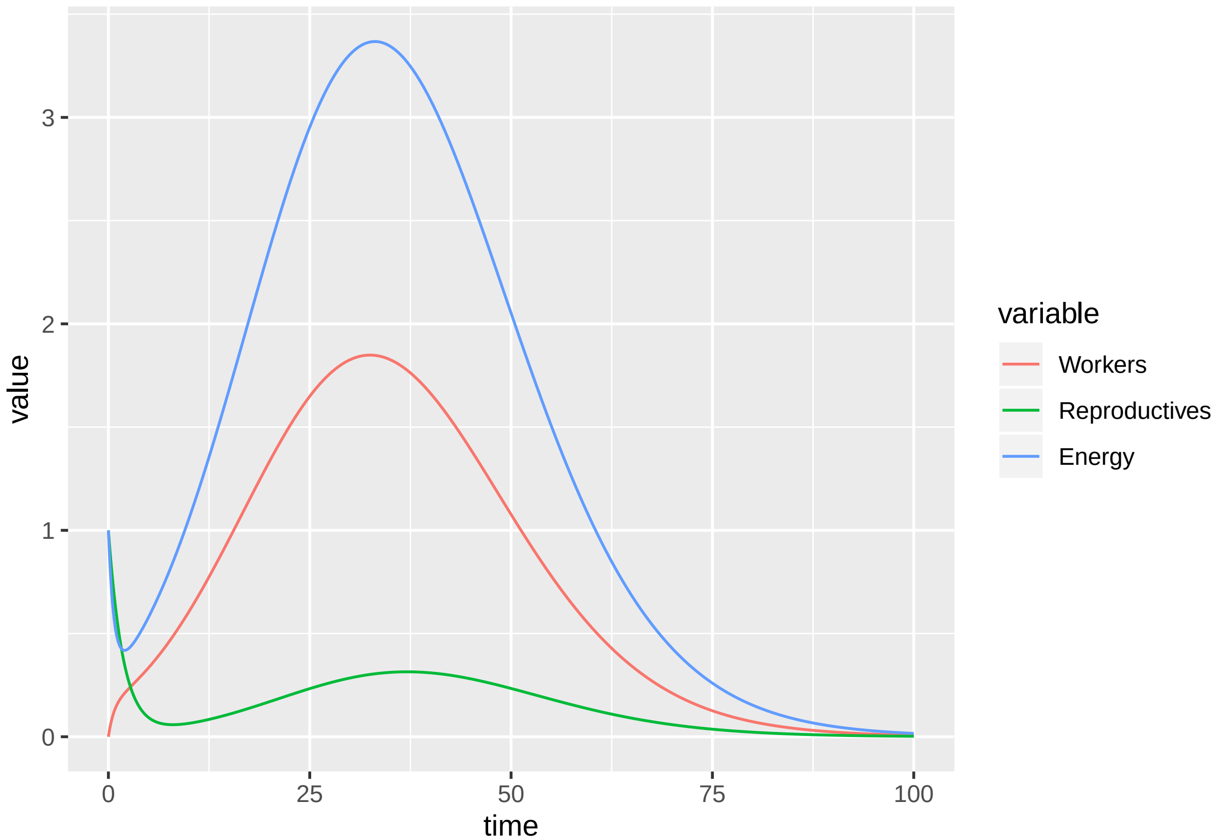
value
3
2
1
0
0
25
50
75
100
time
variable
Workers
Reproductives
Energy


```
##
## [[5]]
```

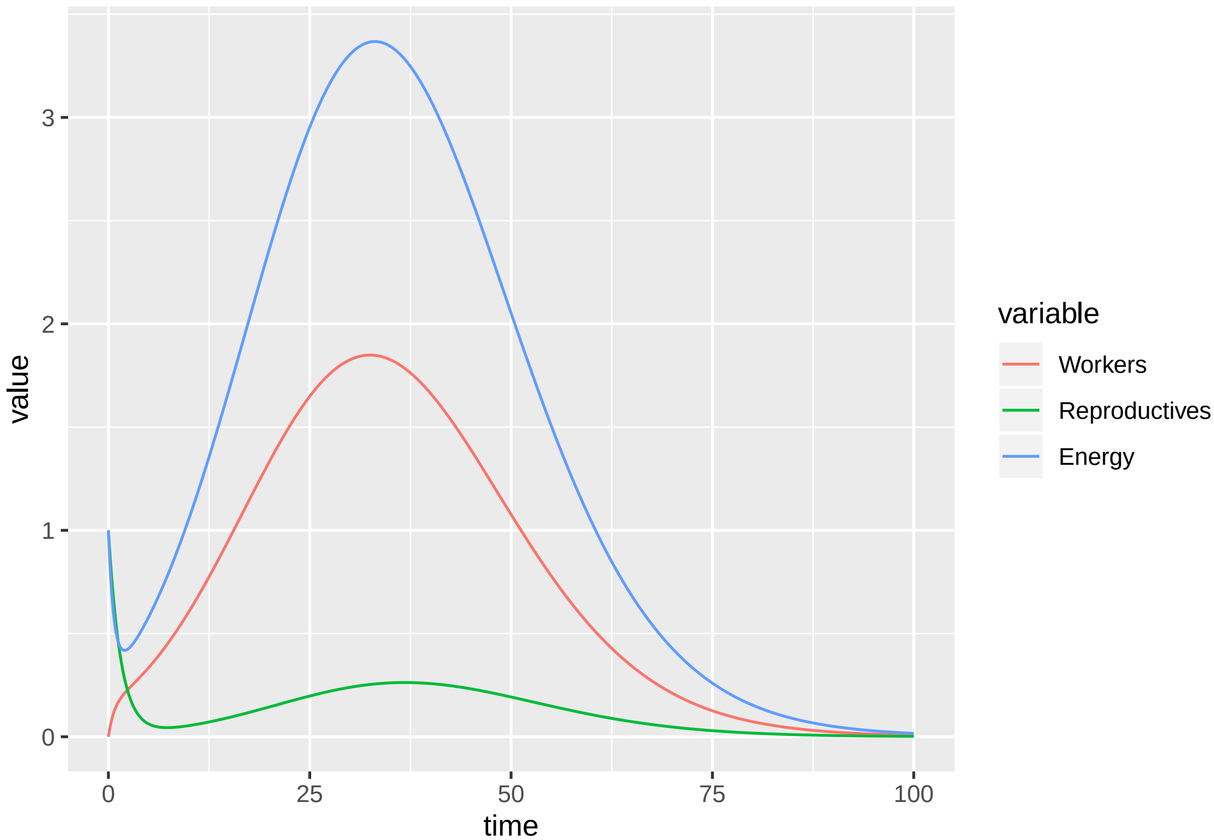


3
2
1
0
value
0
25
50
75
100
time
variable
Workers
Reproductives
Energy


```
##
## [[6]]
```

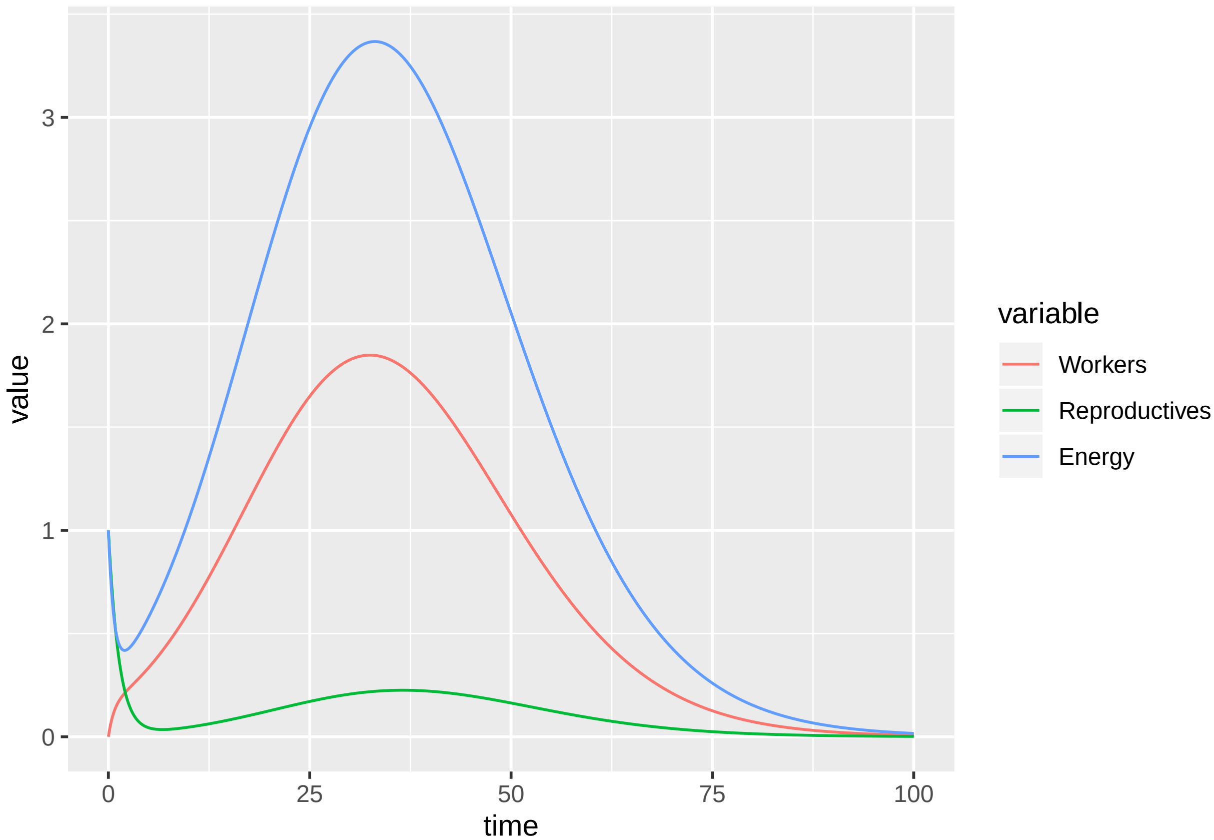

value
3
2
1
0
0
25
50
75
100
time
variable
Workers
Reproductives
Energy

```
##
## [[7]]
```



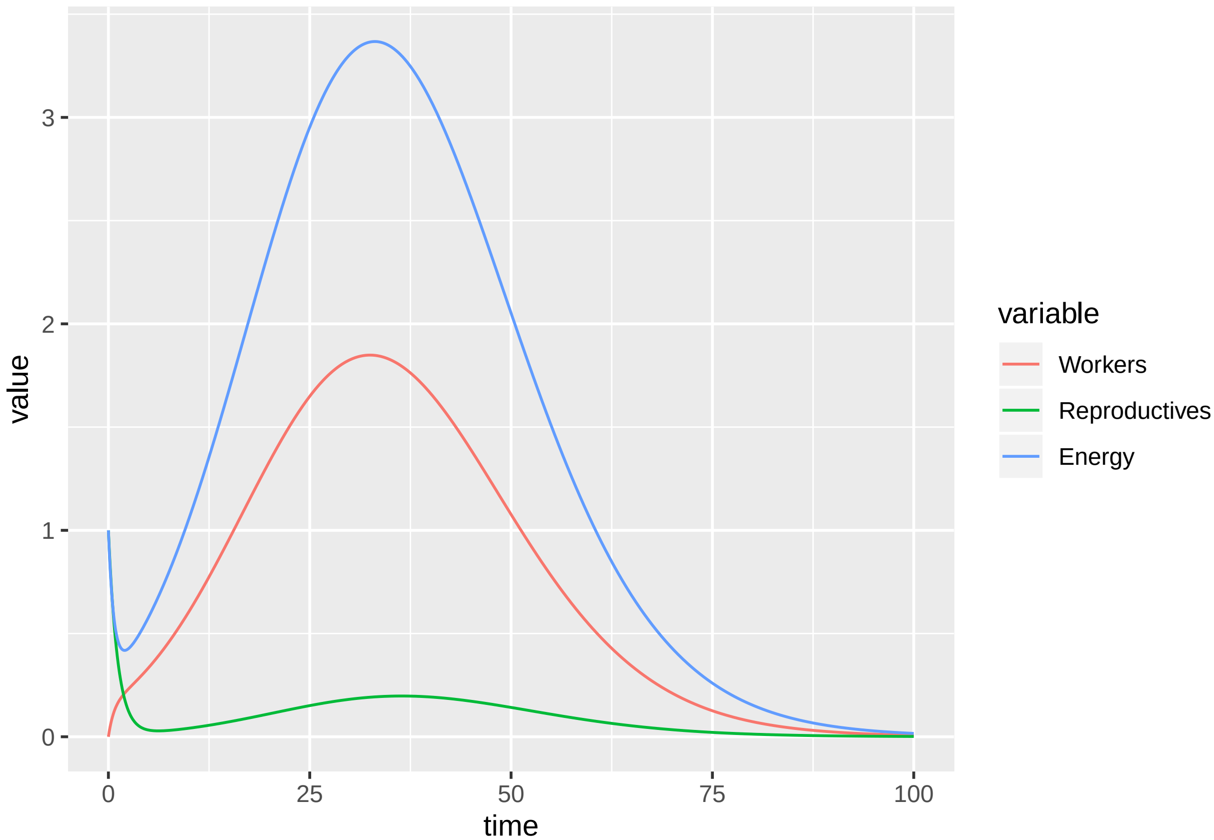
value
3
2
1
0
0
25
50
75
100
time
variable
Workers
Reproductives
Energy


```
##
## [[8]]
```

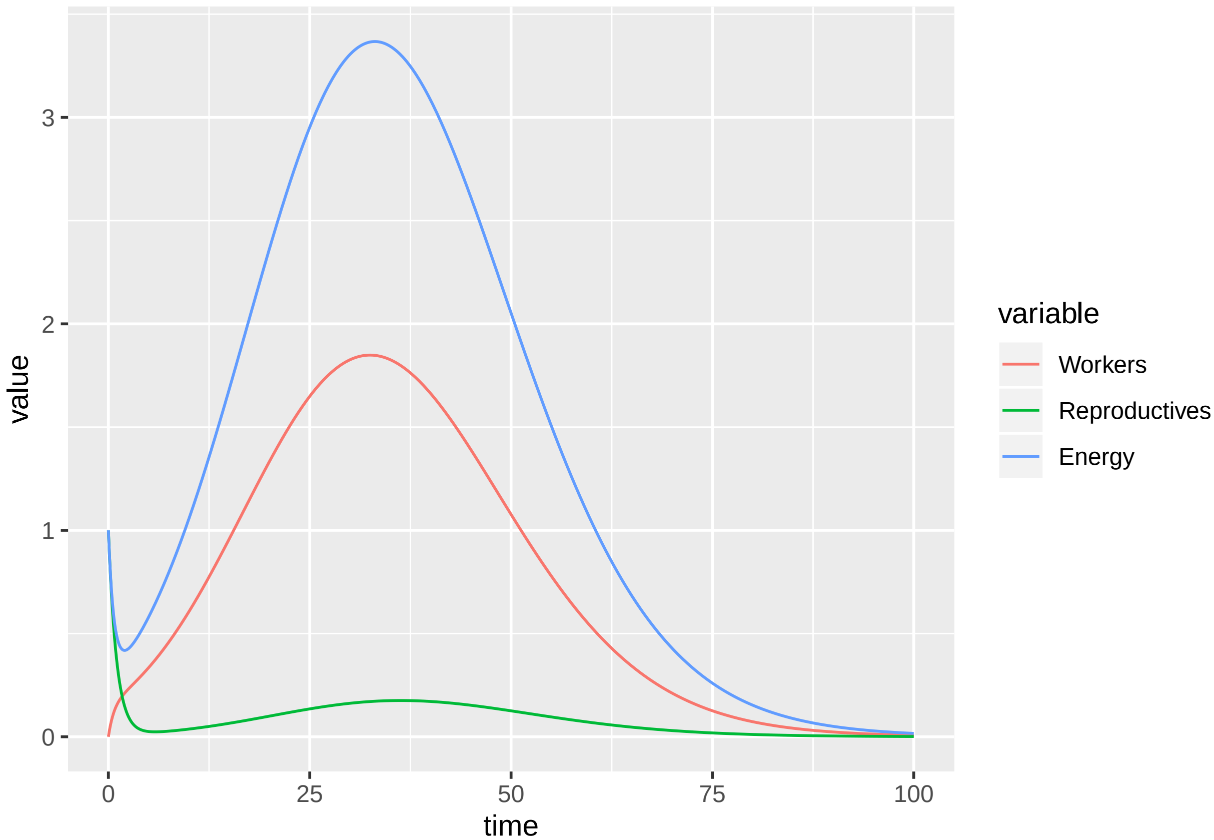


3
2
1
0
value
0
25
50
75
100
time
variable
Workers
Reproductives
Energy


```
##
## [[9]]
```


value
3
2
1
0
time
0
25
50
75
100
variable
Workers
Reproductives
Energy

```
##
## [[10]]
```



3
2
1
0
value
0
25
50
75
100
time
variable
Workers
Reproductives
Energy

```
{r,include=FALSE} # n <- 10 # param.name <- "Df4" # choose
parameter to perturb # param.seq <- seq(.5,.55,length = 10)
# choose range of parameters # Pars <- parameters # Time <-
seq(0, 100, length = n) # param.index <- which(param.name ==
names(Pars)) # out <- list() # for (i in 1:length(param.seq))
#   out[[i]] <- matrix(0, 10, length(state)) #   # for (i in
1:length(param.seq)) { #   # set params #   Pars.loop <- Pars
#   Pars.loop[param.index] <- param.seq[i] #   # converge #
init <- ode(y = state, times = Time, func = systemDE, parms =
Pars.loop) #   # get converged points #   out[[i]] <- ode(init[n,-1],
Time, systemDE, Pars.loop)[,-1] # } #  # range.lim <- lapply(out,
function(x) apply(x, 2, range)) # range.lim <- apply(do.call("rbind",
range.lim), 2, range) # plot.variable <- "R" # choose which
variable to show # plot(0, 0, pch = "", xlab = param.name,
ylab = plot.variable) #      #xlim = range(param.seq), ylim =
range.lim[,plot.variable]) #  # for (i in 1:length(param.seq))
{ #   points(rep(param.seq[i], n), out[[i]][,plot.variable]) #
} #
```