# Assignment5

Richard McCormick

February 27th, 2022

Our goal for this assignment is to understand the dynamics of a vector-borne disease that has a single primary host (human) and a single vector species (mosquito). Consider the following set of equations:

Mosquitoes:

$$\frac{dX_M}{dt} = \mu_M N_M - rT_{MH} X_M Y_H - \mu_M X_M$$
$$\frac{dY_M}{dt} = rT_{MH} X_M Y_H - \mu_M Y_M$$

Humans:

$$\frac{dX_H}{dt} = \mu_H N_H - rT_{HM} X_H Y_M - \mu_H X_H$$
$$\frac{dY_H}{dt} = rT_{HM} X_H Y_M - (\gamma_H + \mu_H) Y_H$$
$$\frac{dZ_H}{dt} = \gamma_H Y_H - \mu_H Z_H$$

## Task 1 (10 points)

Imagine you were using this model in a manuscript to be submitted to a scientific journal. In a brief paragraph (max 300 words), describe the biological processes and assumptions that are represented by these equations. Within your paragraph (and not as a list), define each parameter verbally. This is good practice for explaining equations to a broad audience.

**This is an equational model of a disease which is transferable between mosquiotes and humans. This model uses an SI model for Mosquitoes, which is to say that infected mosquitoes will never recover from the disease. For humans, an SIR model is used, which assumes that humans are able to recover from the disease.**

**The variable $\mu_{MorH}$ represents the birth rate of mosquitoes (M) or humans (H), and $N_{MorH}$ represents the population size of humans or mosquitoes. $T_{MH}$ represents the rate of transmission from Humans to Mosquitoes, and $T_{HM}$ represents the rate of transmission from Mosquitoes to Humans. $\gamma_H$ represents the average recovery period for humans.**

# Task 2 (25 points)

Use previous code and your mathematical wisdom to computationally solve this set of 5 ordinary differential equations. Use your knowledge and intuition to assign realistic values for the parameters. *Hints*: Use a time unit of day$^{-1}$. Transmission probabilities range 0-1. Assume that the constant mosquito biting rate, $b$, is less than 1 per day, and use this $b$ to calculate $r$.

- Create a figure of the dynamics showing each model class as a different line (use line colors and/or line types to define the classes). Label your axes with proper units. *Hint*: Put your y-axis on a log-10 scale (i.e., use `log10()` to transform your density outputs).
- Run your ODEs long enough to see an **endemic equilibrium**.
- Create a figure legend (on the graph or as text below the graph) to define the line colors/types.

```
library(deSolve)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

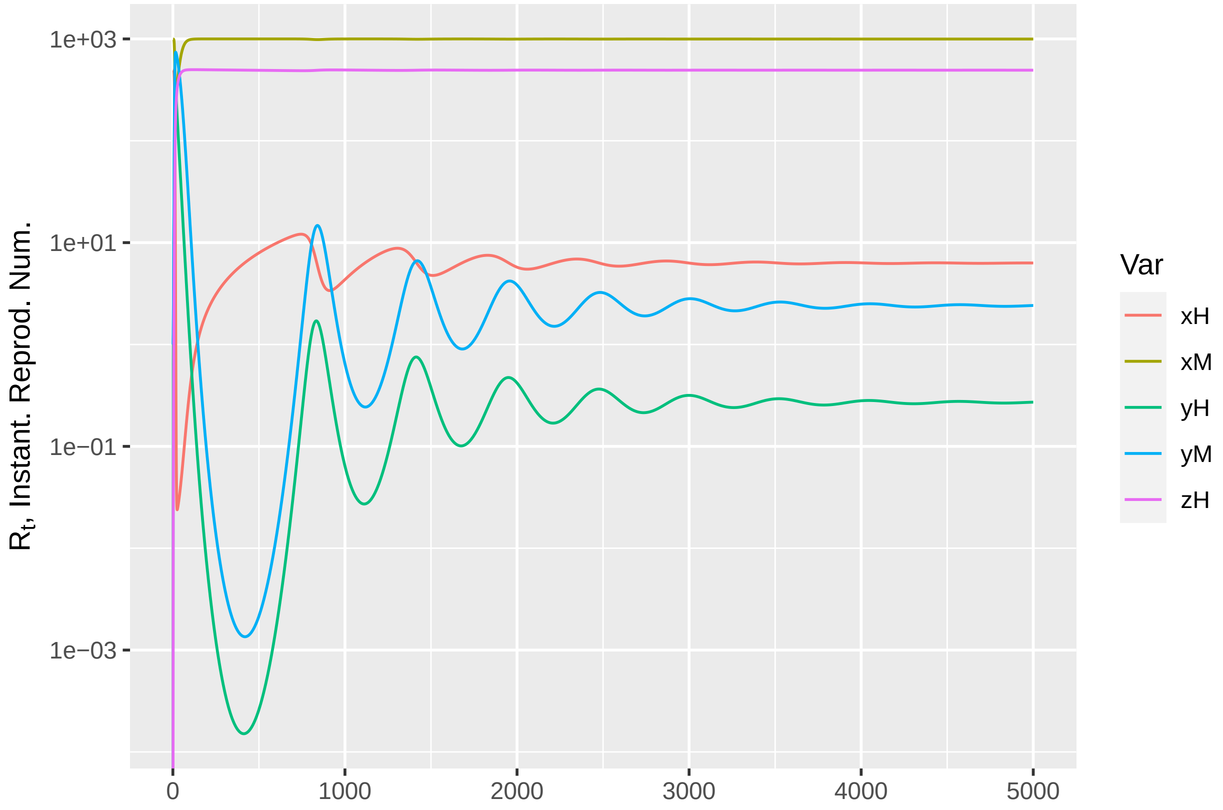
```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(EpiEstim)

muM = 1/7
muH = 1/(70*365)
nM = 1000
nH = 500
b = 0.85

r = b/nH

tMH = 0.75
tHM = 0.75
gammaH = 1/14

params = c(muM = muM,
           muH = muH,
           nM = nM,
           tMH = tMH,
           tHM = tHM,
           r = r,
           gammaH = gammaH)
```

```r
Human_Mosquito_ODE = function(t, y, params){
  with(as.list(c(y, params)), {
    dydt = rep(0, 5)

    # Mosquito Equations
    dydt[1] = (muM * nM) - (r * tMH * y[1] * y[4]) - (muM * y[1])
    dydt[2] = (r * tMH * y[1] * y[4]) - (muM * y[2])

    # Human Equations
    dydt[3] = (muH * nH) - (r * tHM * y[3] * y[2]) - (muH * y[3])
    dydt[4] = (r * tHM * y[3] * y[2]) - ((gammaH + muH) * y[4])
    dydt[5] = (gammaH * y[4]) - (muH * y[5])

    return(list(dydt))

  })
}

#Initial conditions
xM = nM - 1
yM = 1
xH = nH
yH = 0
zH = 0

# Store in a vector
inits = c(xM, yM, xH, yH, zH)

# Each time step that will be solved
time_ode = seq(1, 5000, by = 1)

# Run ODE
out = ode(y = inits,
          times = time_ode,
          func = Human_Mosquito_ODE,
          method="ode45", # Runge-Kutta 4-5 method
          parms = params)

# Object 'out' is a matrix
# Specify the column names, convert to data frame:
colnames(out) = c("time", "xM", "yM", "xH", "yH", "zH")
out = data.frame(out)

long_out =
  out %>%
  pivot_longer(cols=2:6,
               names_to = "Var",
               values_to = "Vals")

ggplot(long_out) +
  geom_path(aes(x = time, y = Vals, color = Var)) +
  labs(x="", y=expression(R[t]*", Instant. Reprod. Num.")) +
  scale_y_continuous(trans="log10")
```

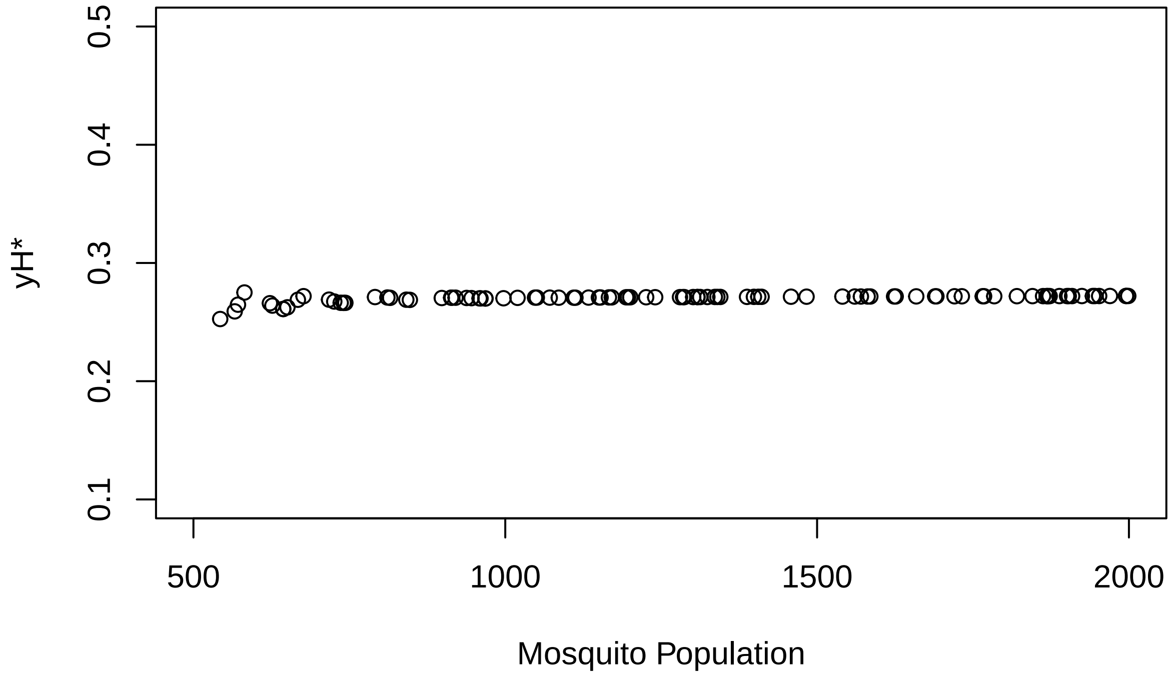
```
## Warning: Transformation introduced infinite values in continuous y-axis
```



## Task 3 (15 points)

As noted above, these equations reach an endemic equilibrium. Create a plot that demonstrates how the equilibrium number of infected humans (${Y_H}^*$) depends on the size of the mosquito population ($N_M$). Keep all other parameters constant. You need to think about how to extract ${Y_H}^*$ from your model output (we are not calculating it analytically). *Hint*: You should not see oscillations in this figure. If you do, you need to run your model for more time, to reach the endemic equilibrium across all your values of $N_M$.

```
runs = 100

time_ode = seq(1, 7000, by=1)

plot(x=NA,
   y=NA,
   xlim = c(500, 2000),
   ylim = c(0.1, 0.5),
   xlab = "Mosquito Population",
   ylab = "yH*")

nM_thing = runif(runs, min=500, max=2000)

for (i in 1:runs)
{
  nM = nM_thing[i]
  params = c(muM = muM,
             muH = muH,
             nM = nM,
             tMH = tMH,
             tHM = tHM,
             r = r,
             gammaH = gammaH)

  out = ode(y = inits,
          times = time_ode,
          func = Human_Mosquito_ODE,
          method="ode45", # Runge-Kutta 4-5 method
          parms = params)

  colnames(out) = c("time", "xM", "yM", "xH", "yH", "zH")
  out = data.frame(out)

  points(nM, tail(out$yH, n=1), col="black")
}
```


yH*
Mosquito Population