

Mathematics 4MB3/6MB3 Mathematical Biology
2018 ASSIGNMENT 2

Group Name: The Infective Collective

Group Members: Aurora Basinski-Ferris, Michael Chong, Daniel Park, Daniel Presta

This assignment was due in class on **Monday 5 February 2018 at 11:30am.**

1 Plot P&I mortality in Philadelphia in 1918

- (a) Confirm that you have received this data file by e-mail:

pim_us_phila_city_1918_dy.csv

This plain text comma-separated-value file can be examined (if you wish) using any plain text editor, such as Emacs.

- (b) Read the data into a data frame in R, using the `read.csv()` function. For example, the following chunk of R code should work:

```
datafile <- "pim_us_phila_city_1918_dy.csv"
philadata <- read.csv(datafile)
philadata$date <- as.Date(philadata$date)
```

The purpose of the last line of code above is to ensure that R encodes character strings such as "1918-10-15" as dates.

- (c) Reproduce the Philadelphia 1918 P&I plot:

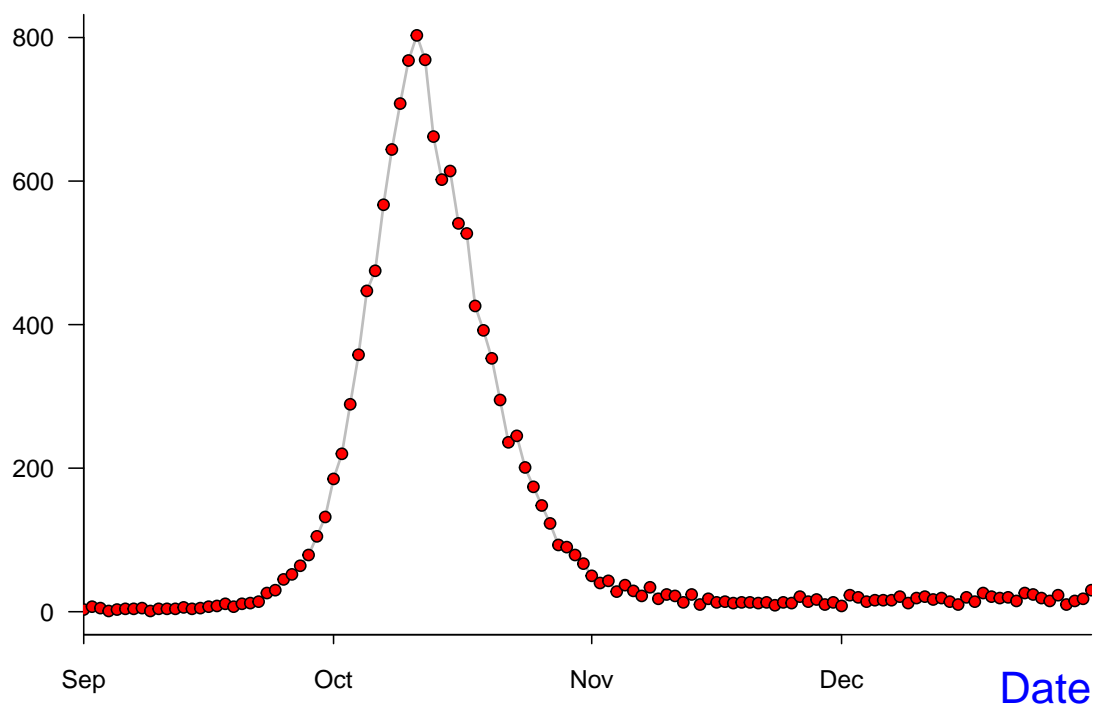
```
Solution. ## first make the box with no annotation or curves
plot(philadata$date, philadata$pim, type="n",
     bty="L", # no upper or right box lines
     ylim=c(0,800), # axis limits
     yaxp=c(0,800,4), #first two numbers is coordinates of
                      #extreme tick marks, third number is num. of marks
     xaxt='n', #supress x ticks and labels
     xlab="",
     ann=FALSE, # no axis annotation (i.e., no title or axis labels)
     xaxs="i", #first tick on x axis is the y axis
     las=1 # axis label style: always horizontal
)
month <- c(9,10,11,12) #want sept, oct, nov, dec labels
ticks <- seq(philadata$date[1],
             philadata$date[length(philadata$date)], by="months")
```

```
#put ticks where we want them and labels:
axis(1, at = ticks, labels = month.abb[month], tcl = -0.3)

## create blue labels for axes
mtext("Date", side=1, adj=1, line=1.5, font=1, cex=1.75, col="blue")
mtext("P&I Deaths", side=2, at=900,
      line=-4, font=1, las=1, cex=1.75, col="blue")
# at =900 is because it is at around 900 on the y axis plot
# (just above the top which is 800)
# font=1 because normal font (not italics or bold)

## plot data
#putting the grey line (lwd gives line thickness relative to default):
lines(philadata, col="grey", lwd=1.75)
#putting the red points:
points(philadata$date, philadata$pim, pch=21, bg="red")
```

P&I Deaths



□

You'll need to use functions such as `plot()`, `points()` and `lines()`. For a comprehensive list of graphics parameters accepted by these functions, enter `?par` into the Console

pane in RStudio. There are multiple ways to produce a graph exactly like the above, but the following steps work:

- Use `plot()` to draw the box and basic annotation and the grey line. Suppress labels when doing this (*e.g.*, `xlab=""`). The box type is controlled by the `bty` option and the orientation of annotation is controlled by the `las` option.
- Use `points()` to draw the heavy red dots with black borders. The most elegant way to do this is to set the point character type to 21 (`pch=21`) and the point background colour to red (`bg="red"`). Alternatively, you can use `points()` twice (first to draw the red dots and then to draw the black circles around them).
- Use `mtext()` to add the x and y axis labels in the margins of the plot.

2 Estimate \mathcal{R}_0 from the Philadelphia P&I time series

- (a) The observed mortality time series $M(t)$ is certainly not equal to the prevalence $I(t)$ that appears in the SIR model. Suppose, however, that $I(t) = \eta M(t - \tau)$ for all time (where η and τ are constants), *i.e.*, that the mortality curve is exactly a scaled and translated version of the prevalence curve. Prove that if both I and M are growing exactly exponentially over some time period then their exponential rates are identical. Thus, if we compare them during the “exponential phase” on a logarithmic scale, then both curves will be perfectly straight with exactly the same slope.

Solution. Let mortality be denoted by $M(t)$ and prevalence be denoted by $I(t)$. If we assume that both mortality and prevalence grow exponentially, then we can write

$$M(t) = ae^{bt}$$

and

$$I(t) = ge^{ht}$$

for some constants a , b , g , and h . Furthermore, if we assume that $I(t) = \eta M(t - \tau)$, for some η and τ , then we can write

$$ge^{ht} = I(t) = \eta M(t - \tau) = \eta ae^{b(t-\tau)}.$$

Then $ge^{ht} = \eta ae^{-\tau} e^{bt}$, and

$$e^{(h-b)t} = \frac{\eta ae^{-\tau}}{g}.$$

Notice that the RHS of this equation is constant and thus forces $h - b = 0$. That is, $h = b$, and thus both $I(t)$ and $M(t)$ have the same exponential growth rate. \square

- (b) Fit a straight line to the part of the Philadelphia 1918 mortality time series that looks straight on a logarithmic scale (and show your result in a plot). Once you get the hang of it, the easiest way to do this is to use the `lm()` function in \mathbb{R} (`lm` stands for linear model). Note that the simplest way to draw a straight line with given slope and

intercept is with the `abline()` function. If you find `lm()` counter-intuitive to understand then experiment with `abline()` until your eyes tell you that you have discovered a line that provides a good fit.

Solution.

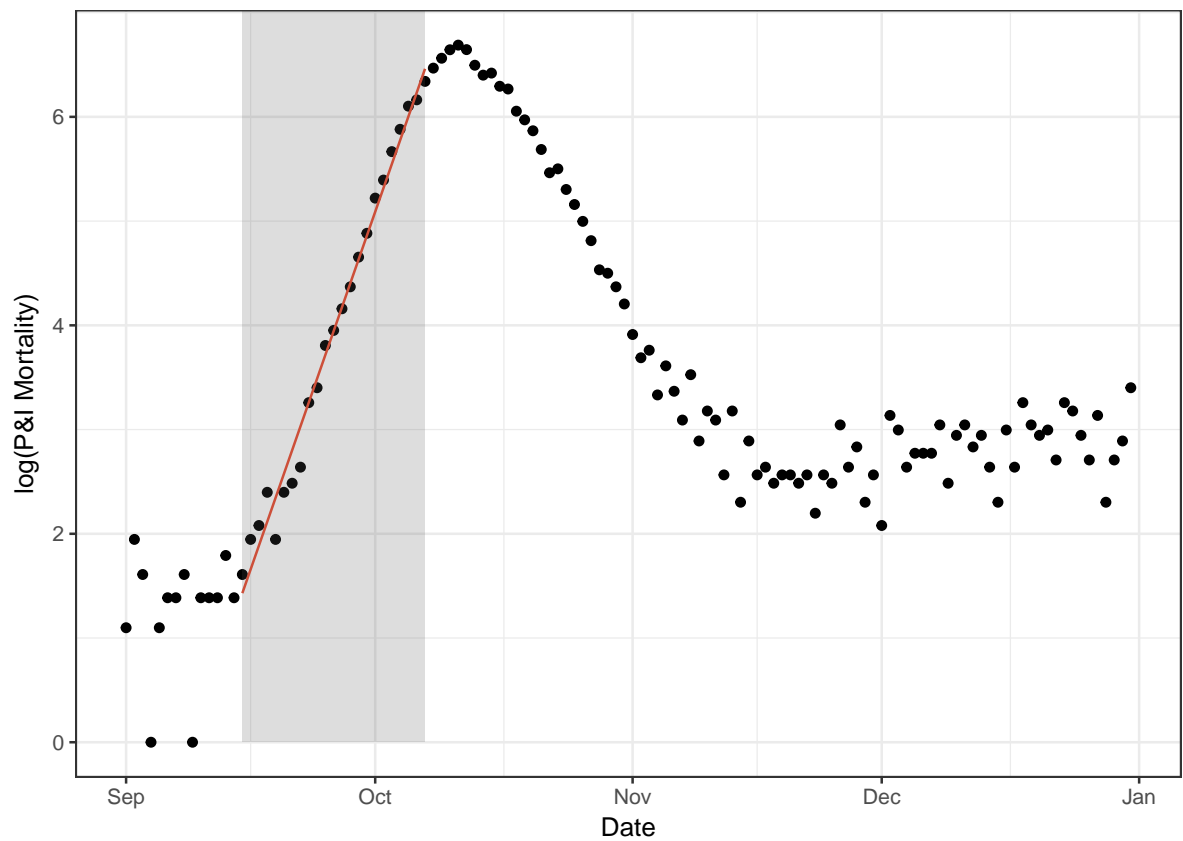
```
library(tidyverse)
library(ggplot2); theme_set(theme_bw())

philadata <- philadata %>%
  mutate(day = as.numeric(date) - min(as.numeric(philadata$date)))
#R stored dates as -18000 when numeric. transformed so first date sept 1918 is 0
x <- philadata %>%
  filter(date <= "1918-10-7" & date >= "1918-09-15")
#filter data in linear region

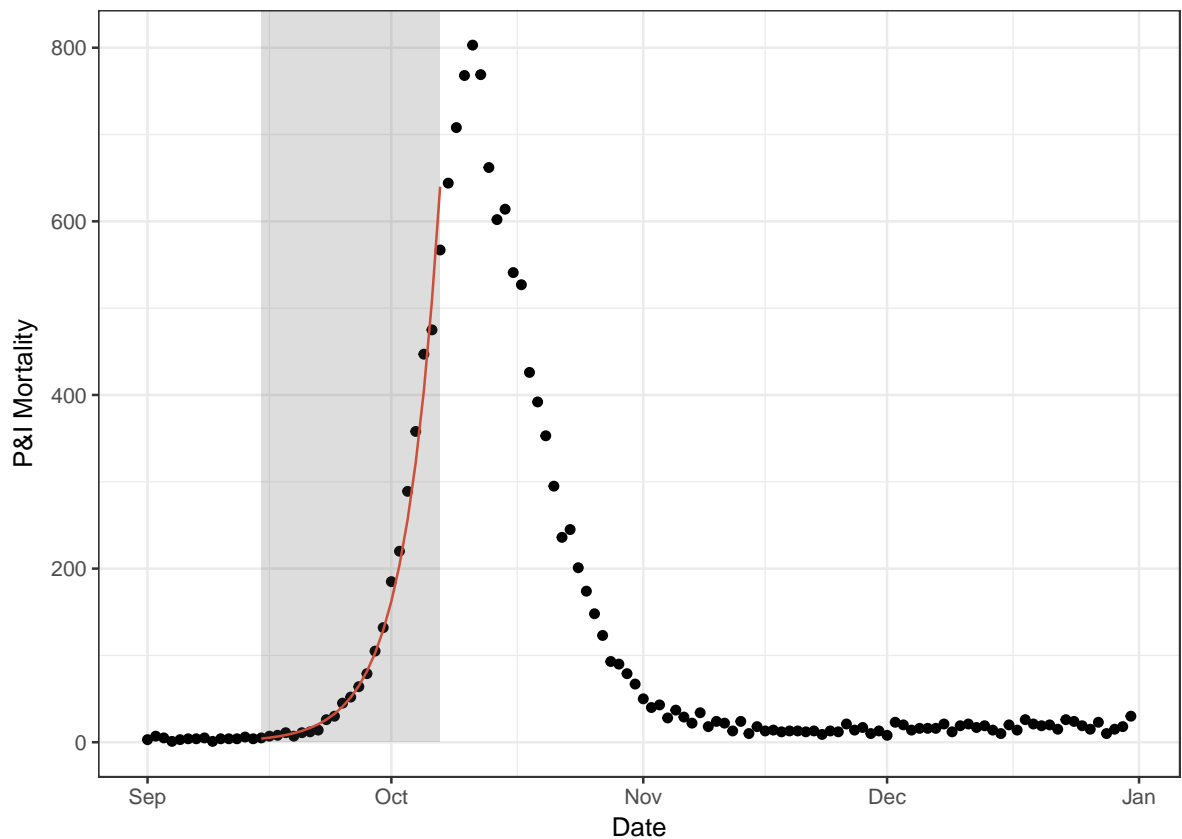
#generate lm and give slope in base e
lm.base.e <- lm(log(pim)~day, data= x)

#add column to x data with pts expected by our model above:
x <- x %>%
  mutate(expected= exp(lm.base.e$coefficients[1])*
    exp(lm.base.e$coefficients[2]*day))

#plot on log scale:
ggplot(philadata, aes(x=date, y=log(pim))) +
  geom_point() +
  annotate(geom="rect", xmin = as.Date("1918-10-7"),
    xmax = as.Date("1918-09-15"), ymin =0, ymax=Inf,
    alpha=0.2) +
  labs(x="Date", y="log(P&I Mortality)") +
  geom_line(data=x, aes(x=date, y=log(expected)), color="tomato3")
```



```
#plot on normal scale and show how model fits:
ggplot(philadata, aes(x=date, y=pim)) +
  geom_point() +
  annotate(geom="rect", xmin = as.Date("1918-10-7"),
          xmax = as.Date("1918-09-15"), ymin = 0, ymax=Inf,
          alpha=0.2) +
  labs(x="Date", y="P&I Mortality") +
  geom_line(data=x, aes(x=date, y=expected), color="tomato3")
```



To determine coefficients, we restricted the data to the portion in which the semi-log plot looks approximately linear, and fit a linear model using the `lm()` function to the log-transformed data. The slope and intercept of this fit is given below.

$$\begin{array}{cc} \text{slope} & \text{intercept} \\ \hline 0.2286857 & -1.7714926 \end{array}$$

□

- (c) How is the slope of your fitted line related to the parameters of the SIR model? (*Hint:* When I is small, $S \simeq 1$.) Why do you need an independent measure of the mean infectious period to estimate \mathcal{R}_0 ? If the mean infectious period is 4 days, what is your estimate of \mathcal{R}_0 ?

Solution. We recall that the SIR model is given by Equation 1.

$$\begin{aligned} \frac{dS}{dt} &= -\beta SI \\ \frac{dI}{dt} &= \beta SI - \gamma I \\ \frac{dR}{dt} &= \gamma I \end{aligned} \tag{1}$$

Given that we are examining data that starts when I is small (and then we have a constant exponential growth rate of 0.2287), we can use the assumption that $S \simeq 1$. Thus, this yields the equation $\frac{dI}{dt} \simeq \beta I - \gamma I$. Solving this, we have Equation 2

$$I = ce^{(\beta-\gamma)t} \quad (2)$$

Following from the answer in Question 2 Part a, we know that the slope of 0.2287 of our fitted mortality curve in Part b is equal to the $\beta - \gamma$ slope we would have if we took the log of Equation 2.

We recall from the SIR model that the constant \mathcal{R}_0 is given by the product of the transmission rate and the mean infection period. Equivalently, we have that $\mathcal{R}_0 = \frac{\beta}{\gamma}$. Thus, in order to establish a value for \mathcal{R}_0 , we need either γ or β , as then we can solve for the other variable using that $\beta - \gamma = 0.2287$. It is much more logical to look for an independent measure of γ , as the mean infectious period given by $\frac{1}{\gamma}$ is easier to infer from data than the transmission rate.


If in our case, the mean infectious period ($\frac{1}{\gamma}$) is 4 days, then we know that $\gamma = \frac{1}{4}$. Thus, we have that $\beta = 0.4787$. From this information, we can solve for $\mathcal{R}_0 = 0.4787 * 4 = 1.9148$. □

3 Fit the basic SIR model to the Philadelphia P&I time series

- (a) Install the "deSolve" package. This is done by typing the following command in the Console pane of RStudio:

```
install.packages("deSolve")
```

You will then be prompted to choose a mirror site from which to download the package. It doesn't matter which mirror you choose, but choosing a site in Ontario might save a fraction of a second. *Note:* This is a one-time operation. You do not want an `install.packages()` command inside your solutions code.

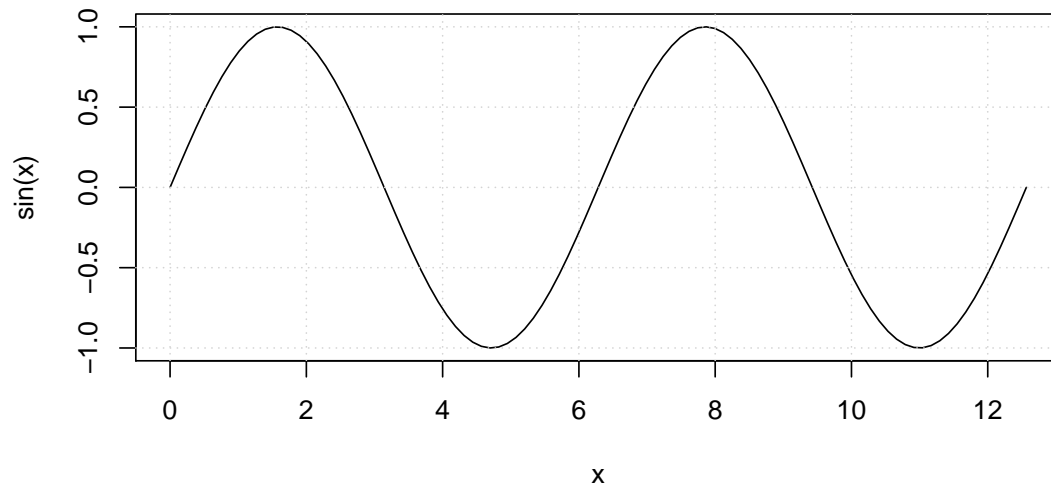
- (b) Write an  function that plots the solution $I(t)$ of the SIR model for given parameter values (\mathcal{R}_0 and $1/\gamma$) and given initial conditions (S_0, I_0). Use the `ode()` function in the `deSolve` package. A few hints:

- Your code will first need to load the `deSolve` package:

```
library(deSolve)
```

- As an example of defining a function (without getting involved with a differential equation), here is a code chunk that defines a function to plot a sine curve, and then executes the function. Note that the default min and max x values are set in the parameter list of the function definition, but the max x value is changed when the function is executed:

```
plot.sine <- function( xmin=0, xmax=2*pi ) {
  x <- seq(xmin,xmax,length=100)
  plot(x, sin(x), typ="l")
  grid() # add a light grey grid
}
plot.sine(xmax=4*pi)
```



- Here's another example. This time we first define the vector field for a differential equation. We then use this function inside another function that plots the solution of the associated differential equation. To understand the construction, you can, as usual, study the help page for the calling function (`?ode` in this case), but the most important issues are the following.

One of the arguments of the `ode()` function is the function that evaluates the vector field at the current time. To avoid confusion, choose the arguments of your vector field function to be `t`, `vars` and `parms` (in that order):

- t** The current time, which will be used within the vector field function if the system is non-autonomous.
- vars** A named vector of the variables in the system (*e.g.*, S , I). The variables, as named vector passed to this function, are used in the code that defines the vector field within the function.
- parms** A named vector of the parameters of the system (*e.g.*, β , γ). It is convenient—but not necessary—to specify default values for the parameters.

It is strongly recommended that you follow exactly the style below when defining vector fields for differential equations that you wish to solve with the `ode()` function.

In particular, the construction “`with(as.list(c(parms,vars)), ...)`” makes the variables and parameters visible within the section of code between the braces (`{...}`) without having to refer to the vectors or lists in which they are stored. For example, the code would be much harder to read if each instance of `x` were replaced by `vars$x` and each instance of `beta` were replaced by `parms$beta`; this issue becomes extremely important for complicated vector fields.

```
## Vector Field for SI model
SI.vector.field <- function(t, vars, parms=c(beta=2,gamma=1)) {
  with(as.list(c(parms, vars)), {
    dx <- -beta*x*y # dS/dt
    dy <- beta*x*y   # dI/dt
    vec.fld <- c(dx=dx, dy=dy)
    return(list(vec.fld)) # ode() requires a list
  })
}
```

The following function plots a single solution of the ODE for a given initial condition (`ic`), integration time (`tmax`) and times at which the state is to be returned (`times`). The vector field function is passed as the `func` argument and the parameter vector is passed as the `parms` argument. If further arguments are given, they are passed to the `lines()` function that draws the solution.

```
## Draw solution
draw.soln <- function(ic=c(x=1,y=0), tmax=1,
                      times=seq(0,tmax,by=tmax/500),
                      func, parms,
                      col="blue",
                      ... ) {
  soln <- ode(ic, times, func, parms)
  lines(times, soln[, "y"], col=col, lwd=3, ... )
}
```

Note here that the call to the `ode()` function gives the arguments in the default order so they are interpreted correctly. If we wished to write the arguments in a different order then we would have to be explicit about which argument is which. For example, if we wanted to list the initial conditions last for some deep reason then we would have to write:

```
soln <- ode(times=times, func=func, parms=parms, y=ic)
```

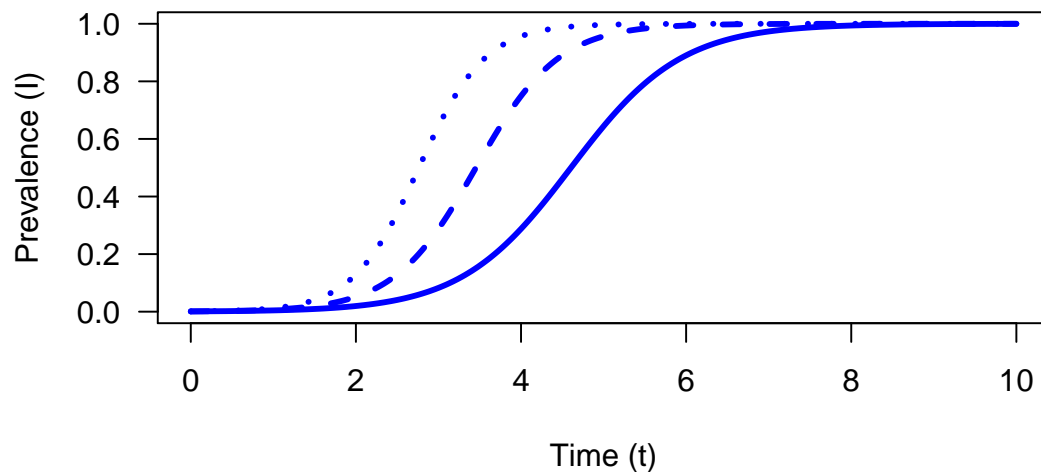
We can now use our `draw.soln()` function to plot a few solutions of the SI model.

```
## Plot solutions of the SI model
tmax <- 10 # end time for numerical integration of the ODE
## draw box for plot:
```

```

plot(0,0,xlim=c(0,tmax),ylim=c(0,1),
     type="n",xlab="Time (t)",ylab="Prevalence (I)",las=1)
## initial conditions:
I0 <- 0.001
S0 <- 1 - I0
## draw solutions for several values of parameter beta:
betavals <- c(1.5,2,2.5)
for (i in 1:length(betavals)) {
  draw.soln(ic=c(x=S0,y=I0), tmax=tmax,
            func=SI.vector.field,
            parms=c(beta=betavals[i],gamma=1),
            lty=i # use a different line style for each solution
  )
}

```



Solution. First, we must determine values for β and γ , since the basic SIR model that we have studied in class has parameters β and γ . We can do so by following the method described in problem 2 (c). We observe that γ is the reciprocal of the infectious period, and that $\beta = \mathcal{R}_0\gamma$. If we suppose that $I_0 = 10^{-3}$ and $S_0 = 1 - I_0$, we can follow the template given in the question to graph solutions for a mean infectious period of 4 days, and for various \mathcal{R}_0 values. We begin by defining a vector field for the SIR model:

```

## Vector Field for SIR model
SI.vector.field <- function(t, vars, parms=c(beta=2,gamma=1)) {
  with(as.list(c(parms, vars)), {
    inf <- beta*x*y ## infection

```

```

        return(list(c(dx=-inf, dy=inf-gamma*y)))
    })
}

```

Then, we can define a function that draws the solution to the SIR model given parameters, initial values, and the maximum time step:

```

draw.sir <- function(S0,
                    I0,
                    R0,
                    inf.period,
                    tmax, ## max time
                    ...) {
  ## draw solution curve (prevalence)
  draw.soln(ic=c(x=S0, y=I0),
            tmax=tmax,
            func=SIR.vector.field,
            parms=c(beta=R0/inf.period,
                    gamma=1/inf.period),
            ...)
}

```

□

- (c) For $I_0 = 10^{-3}$ and $S_0 = 1 - I_0$, plot the solutions of the SIR model assuming $1/\gamma = 4$ days and $\mathcal{R}_0 \in \{1.2, 1.5, 1.8, 2, 3, 4\}$. Use the `legend()` command to make a legend on the plot that shows which curves correspond to which values of \mathcal{R}_0 .

```

Solution. ## setting up initial conditions
## as well as parameters
I0 <- 1e-3
S0 <- 1 - I0
inf.period <- 4
R0 <- c(1.2, 1.5, 1.8, 2, 3, 4)
n <- length(R0)
tmax <- 150

## set up plot window
plot(NA, xlim=c(0,tmax), ylim=c(0,0.4),
     xlab="Time (days)", ylab="Prevalence")
## actually plot curves
## Map is a fancy way of avoiding loops
SIR.plot <- Map(draw.sir, S0=S0, I0=I0, R0=R0, inf.period=inf.period,

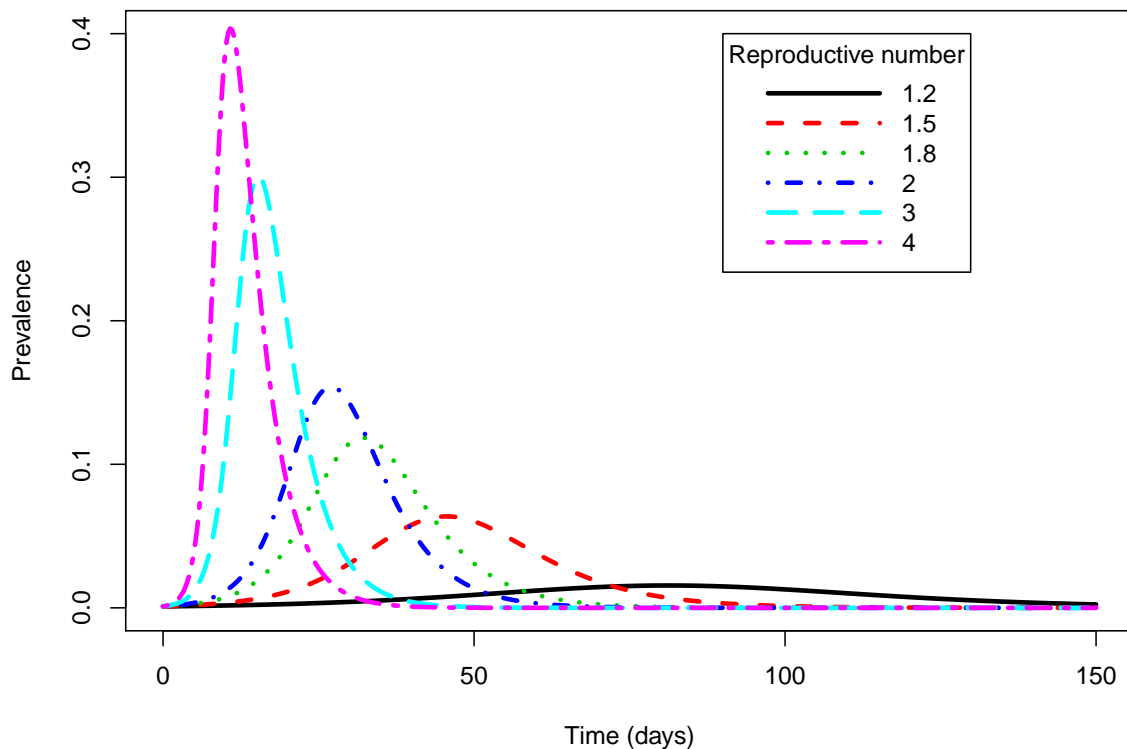
```

```

lty=1:n, col=1:n,
tmax=tmax
)

legend(
  x=90, y=0.4,
  legend=R0, lty=1:n, col=1:n,
  lwd=3, title="Reproductive number", seg.len=5
)

```



□

- (d) By trial and error, find values of \mathcal{R}_0 and γ that yield a solution of the SIR model that fits the Philadelphia P&I times series reasonably well. You can assess the quality of fit using the Euclidean distance between the model solution and the data. (*Note:* The trial and error approach is a valuable exercise, but not a suggestion of a method you would really use in practice. We'll discuss better methods for fitting ODE models to data later.)

Solution. In section 2 (a), it was assumed that mortality and prevalence have the following relationship:

$$I(t) = \eta M(t - \tau).$$

Rearranging, we have

$$M(t) = \frac{1}{\eta} I(t + \tau)$$

So we will have to run the model for $t + \tau$ time steps to compare and estimate η , τ , $S(0)$, $I(0)$, \mathcal{R}_0 , and infectious period.

Here is the function that will return a data frame whose columns are date and expected daily mortality, given parameters and initial conditions:

```
simulate_mortality <- function(S0,
                                I0,
                                R0,
                                inf.period,
                                eta,
                                tau,
                                tmax=nrow(philadata)) {
  ## first we want to solve ode
  soln <- as.data.frame(ode(
    y=c(x=S0, y=I0),
    times=1:(tmax+tau),
    func=SIR.vector.field,
    parms=c(beta=R0/inf.period,
             gamma=1/inf.period)))

  ## then mortality is approximation
  data.frame(
    date=philadata$date,
    pim=tail(soln$y, -tau)/eta
  )
}
```

Now here is the estimate that we found via trial and error:

$$S(0) = 1 - 10^{-7}, I(0) = 10^{-7}, \mathcal{R}_0 = 2.2, \frac{1}{\gamma} = 4, \eta = 0.00025, \tau = 14.$$

Then, we can compare our estimated mortality curve with the given data:

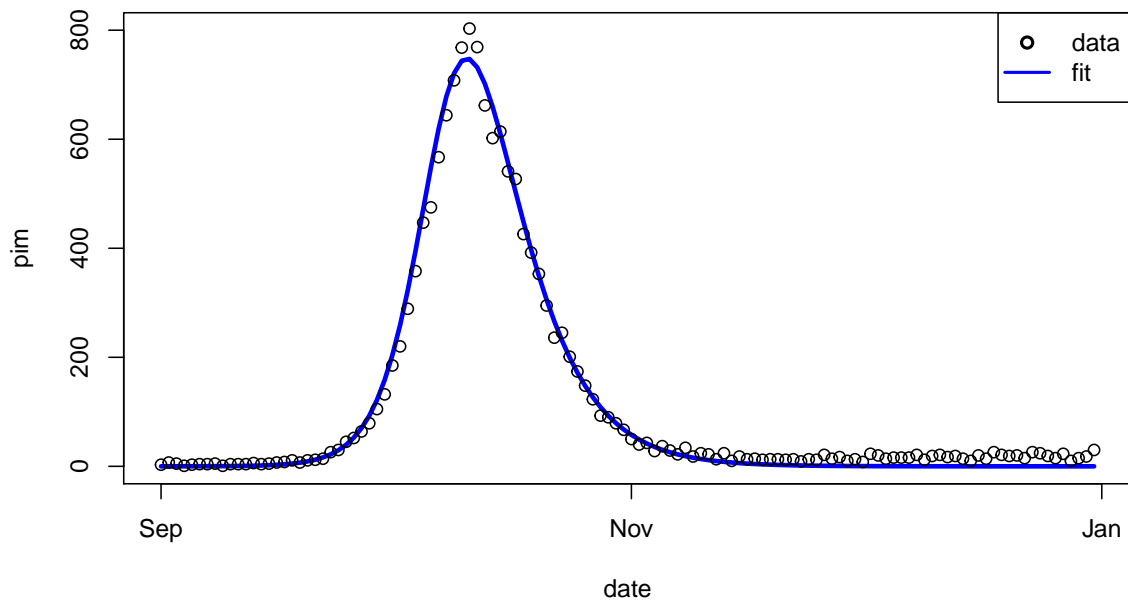
```
## simulating mortality using parameters we estimated
mfit <- simulate_mortality(S0=1-1e-7, I0=1e-7,
                           R0=2.2, inf.period = 4,
                           eta=0.00025, tau=14)

## plotting expected mortality curve
plot(mfit, type="l", lwd=3, col="blue", ylim=c(0,800))
points(philadata)
```

```

legend(
  "topright",
  legend=c("data", "fit"),
  lty=c(NA, 1),
  pch=c(1, NA),
  col=c("black", "blue"),
  lwd=2
)

```

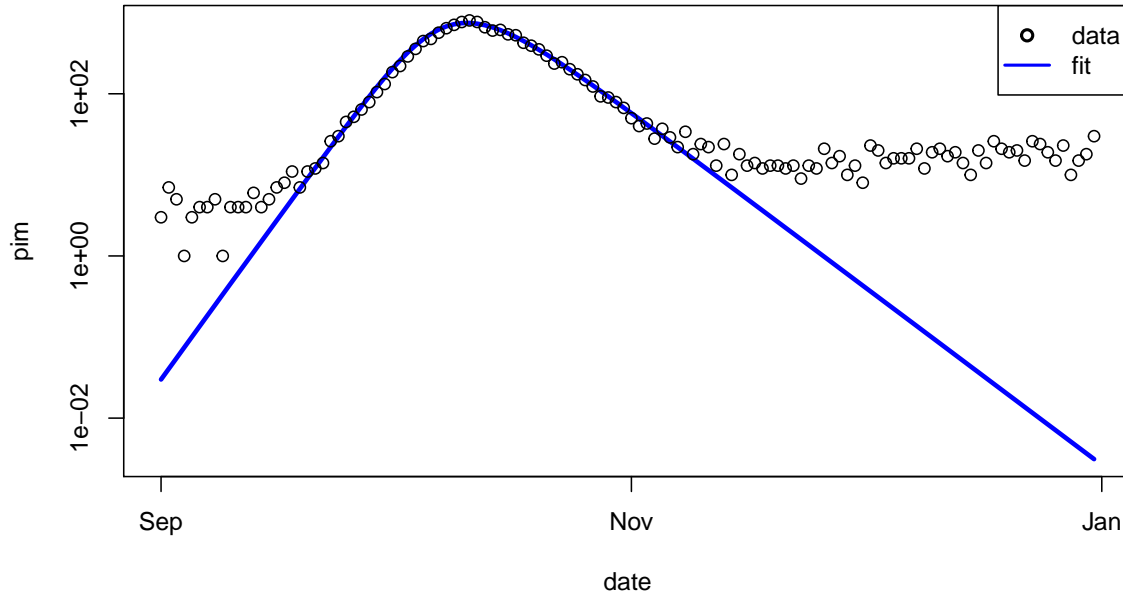


Note that our fit appears to underestimate mortality after November. We can take a closer look at the difference by plotting the graph on a log scale:

```

plot(mfit, type="l", lwd=3, col="blue", log="y")
points(philadata)
legend(
  "topright",
  legend=c("data", "fit"),
  lty=c(NA, 1),
  pch=c(1, NA),
  col=c("black", "blue"),
  lwd=2
)

```



Notice that we have a very good fit in the middle but not near the two opposite ends. The SIR model predicts that the disease will become extinct after a finite period of time. However, mortality data suggests that the epidemic will continue even after a long period of time. Hence, we can conclude that the SIR model might simply be insufficient to explain the data and will not fit very well even if we tried harder. \square

4 Executive summary for the Public Health Agency

The Public Health Agency of Canada (PHAC) is revising their pandemic plan and has asked your group to summarize what you learned from analyzing the 1918 Philadelphia P&I time series. Besides explaining what inferences you feel you can make from your analysis so far, PHAC wants to know what you would investigate if they were to fund you to continue your work full time for a month. They want a maximum of one page from your group.

Incidentally, you might be interested to know that rumour has it that all of the members of the pandemic planning committee took Math 2C03 at McMaster University between 1980 and 2003, but they all failed. Also, when the chair of the committee was recently asked “What is a differential equation?” he apparently bent over and vomited (it is hard to know quite what to make of this given that PHAC was investigating a norovirus outbreak at the time).

Note: When submitting your assignment solution, it is imperative that the one-page executive summary be printed on its own page. To start a new page in \LaTeX , use the `\newpage` command. Also, as usual, your summary should be in 12 point font. Don’t try to cram in as much as possible. Make that page as clear and concise as you can, so that a public health planner can absorb its content quickly and easily.

The 1918 influenza pandemic was one of the worst influenza epidemics ever recorded. Data from this period are therefore valuable to study because they can provide insight into how similar present-day infectious diseases might spread in a population. In this report, we summarize results from the analysis of 1918 pneumonia and influenza mortality data in Philadelphia, and discuss potential implications for influenza dynamics and future outbreaks.

Using these data, we can characterize key features of the spread of this epidemic. We begin by analyzing the average number of secondary cases caused by a primary case. Knowing this quantity allows us to estimate the total number of infected individuals and to predict how many people would potentially be infected in any future flu epidemics. In particular, we observed that the average number of secondary cases caused by a primary case is linked to how fast an epidemic grows and how long an infection lasts. Hence, if we can estimate these quantities in the early stages of these epidemics, we will be able to better prepare for future epidemics.

For the 1918 flu, we estimate based on the Philadelphia data that a primary case of influenza would have caused approximately 1.9 to 2.3 secondary cases, assuming that the average flu infection lasts four days. From this, we estimate that about 76%-86% of the population would have been ultimately infected by this flu strain. Furthermore, we expect that at the peak of the epidemic, about 14%-20% of the population would have been infected simultaneously, meaning that current hospital capacities are unlikely to be able to handle a similar epidemic. In addition, if another flu of this magnitude were to occur, 47%-56% of the population would need to be vaccinated in order to prevent the disease from growing in the population. That is, if this portion of the population were vaccinated, we would expect the total number of simultaneously infected individuals in the population to decrease, thereby preventing an epidemic.

The investigations performed so far were rather preliminary, and additional analysis would be beneficial. If the PHAC were to continue to fund research on historical flu epidemics, we would pursue additional avenues which could help increase the accuracy of our estimate of the average number of secondary cases caused by a primary case. This improved value would in turn provide more accurate information about final expected epidemic size. One possible way to improve the estimate would be to investigate external factors that might help us better explain the data in the 1918 time series. For example, after investigating the biological processes that the spread of influenza relies upon, we would introduce additional terms into our model to try to represent the disease dynamics in greater detail. Another possible way to improve our estimate of the average number of secondary cases caused by a primary case is through examining other time series from epidemics in different locations which were likely caused by the same strain of influenza. This would help us deduce if our model and estimated values are consistent with the dynamics of influenza in general.

— END OF ASSIGNMENT —

Compile time for this document: February 4, 2018 @ 22:08